Kræsjkurs i MATLAB

Jonas van den Brink
j.v.brink@fys.uio.no

Tekna Student Oslo

28. oktober 2014

Hva er Matlab?

Matriser

$$\begin{pmatrix} 4 & 2 & 0 & 1 \\ 3 & 1 & 3 & 3 \\ 0 & 5 & 1 & 2 \\ 2 & 2 & 3 & 0 \end{pmatrix}$$

Vi kan lage arrays og matriser ved å liste alle elementene

```
A = [1 3 4; 2 3 1; 2 1 -1]
```

$$A = \begin{pmatrix} 1 & 3 & 4 \\ 2 & 3 & 1 \\ 2 & 1 & -1 \end{pmatrix}$$

```
x = [1 3 4 2]
```

$$\mathbf{x} = \begin{pmatrix} 1 & 3 & 4 & 2 \end{pmatrix}$$

```
y = [1; 3; 4; 2]
```

$$\mathbf{y} = \begin{pmatrix} 1 \\ 3 \\ 4 \\ 2 \end{pmatrix}$$

Det finnes også innebygde funksjoner som lager gitte matriser

```
A = zeros(2,4)
```

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

```
A = ones(1,3)
```

$$A = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$$

```
A = eye(3,3)
```

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Det finnes også innebygde funksjoner som lager gitte matriser

```
A = rand(4,4)
```

$$A = \begin{pmatrix} 0.8147 & 0.6324 & 0.9575 & 0.9572 \\ 0.9058 & 0.0975 & 0.9649 & 0.4854 \\ 0.1270 & 0.2785 & 0.1576 & 0.8003 \\ 0.9134 & 0.5469 & 0.9706 & 0.1419 \end{pmatrix}$$

Vi kan behandle bestemte matriseelementer ved å indeksere

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

Vi velger element A_{ij} ved å indeksere `A(i,j)`. Matlab begynner å telle fra 1.

Vi kan behandle bestemte matriseelementer ved å indeksere

```
A = [2 3 1; 0 -4 3; 1 2 -1]
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & -4 & 3 \\ 1 & 2 & -1 \end{pmatrix}$$

- Hva er `A(2,1) = 0`?
- Hva er `A(1,1) + A(2,2) + A(3,3)`?
- Hva gjør kommandoen `A(3,2) = 0`?

Pass på indeksering, det kan fort bli feil

```
A = [2 3 1; 0 -4 1; 3 2 2]
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & -4 & 1 \\ 3 & 2 & 2 \end{pmatrix}$$

```
A(3,2)+A(1,4)
```

$\Rightarrow$ Error using * Inner matrix dimensions must agree.

```
A(1,4)=4
```

$$A = \begin{pmatrix} 2 & 3 & 1 & 4 \\ 0 & -4 & 1 & 0 \\ 3 & 2 & 2 & 0 \end{pmatrix}$$

Matlab tolker operasjoner som vektor- og matriseoperasjoner

```
A = [2, 3, 1; 0, 1, 2; 1, -1, 0]
x = [1; 0; 3]
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 1 & -1 & 0 \end{pmatrix}, \qquad x = \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix}.$$

```
y = A*x
```

$$\mathbf{y} = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 1 & -1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 3 \end{pmatrix} = \begin{pmatrix} 5 \\ 6 \\ 1 \end{pmatrix}.$$

Vi kan også regne komponentvis

```
A = [2, 3, 1; 0, 1, 2]
x = [1, -1, 2; 0, 1, 3]
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} 1 & 1 & 2 \\ 0 & 6 & 3 \end{pmatrix}.$$

```
y = A.*x
```

$$\mathbf{y} = \begin{pmatrix} 2 \times 1 & 3 \times 1 & 1 \times 2 \\ 0 \times 0 & 1 \times 6 & 2 \times 3 \end{pmatrix} = \begin{pmatrix} 2 & 3 & 2 \\ 0 & 6 & 6 \end{pmatrix}.$$

Merk at `*` operasjonen betyr matrise-ganger-vektor, mens `.*` betyr komponentvis multiplikasjon.

Man bør være litt forsiktig med hvilken operator man bruker

```
A = [2 3; 1 0]
```

$$A = \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix}.$$

```
B = A^3
```

$$B = \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 2 & 3 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 20 & 21 \\ 7 & 6 \end{pmatrix}.$$

```
C = A.^3
```

$$C = \begin{pmatrix} 2^3 & 3^3 \\ 1^3 & 0^3 \end{pmatrix} = \begin{pmatrix} 8 & 27 \\ 1 & 0 \end{pmatrix}.$$

Dimensjoner må stemme når vi gjør matriseoperasjoner

Matlab tolker egentlig alle arrays som matriser, og gjør dermed forskjellig på kolonne- og radvektorer.

```
x = [2 3 1 0]
y = [1 1 2 4]
```

$$\mathbf{x} = \begin{pmatrix} 2 & 3 & 1 & 0 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 1 & 1 & 2 & 4 \end{pmatrix}.$$

```
x*y
```

$\Rightarrow$ Error using * Inner matrix dimensions must agree.

Vi kan ikke ta matrisproduktet av to radvektorer.

Dimensjoner må stemme når vi gjør matriseoperasjoner

For å ta matriseproduktet av to vektorer, må vi ha en radvektor og en kolonnevektor.

```
x = [2 3 1 0]
y = [1; 1; 2; 4]
```

$$\mathbf{x} = \begin{pmatrix} 2 & 3 & 1 & 0 \end{pmatrix}, \qquad \mathbf{y} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 4 \end{pmatrix}.$$

```
x*y
```

$\Rightarrow$ 7.

Matriser kan enkelt transponeres

```
A = [2 3 1; 0 1 2; 3 0 1]
B = A'
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 3 & 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 2 & 0 & 3 \\ 3 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

Vi kan også transponere vektorer

```
x = [2 3 1]
y = x'
```

$$\mathbf{x} = \begin{pmatrix} 2 & 3 & 1 \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix}.$$

Vi kan regne ut determinanten av en matrise og radredusere den

```
A = [2 3 1; 0 1 2; 3 0 1];
det(A)
```

$\Rightarrow \quad 17.$

```
A = [3 3 1; 6 1 2; 3 2 1]
B = rref(A)
```

$$A = \begin{pmatrix} 3 & 3 & 1 \\ 6 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 1.0000 & 0 & 0.3333 \\ 0 & 1.000 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Matlab kan finne den inverse av en matrise

```
A = [2 3 1; 0 1 2; 3 0 1]
B = inv(A)
```

$$A = \begin{pmatrix} 2 & 3 & 1 \\ 0 & 1 & 2 \\ 3 & 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} 0.0588 & -0.1765 & 0.2941 \\ 0.3529 & -0.0588 & -0.2353 \\ -0.1765 & 0.5294 & 0.1176 \end{pmatrix}.$$

Vi kan teste ved å regne ut AA^{-1}

```
C = A*inv(A)
```

$$C = \begin{pmatrix} 1.0000 & 0 & 0.0000 \\ 0 & 1.0000 & 0 \\ 0 & 0.0000 & 1.0000 \end{pmatrix}.$$

Vi ser at C er identitetsmatrisen. Merk at utksriften blir litt rar fordi matlab regner numerisk.

Vi kan nå løse matriseligninger i Matlab

Ønsker å løse en ligning av typen

$$A\mathbf{x} = \mathbf{b}.$$

Løsningen vil være gitt ved

$$\mathbf{x} = A^{-1}\mathbf{b}.$$

```
A = [2 3; 0 -1]
b = [2; 1]
x = inv(A)*b
```

$$A = \begin{pmatrix} 2 & 3 \\ 0 & -1 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \quad \Rightarrow \quad \mathbf{x} = \begin{pmatrix} 2.5 \\ -1 \end{pmatrix}.$$

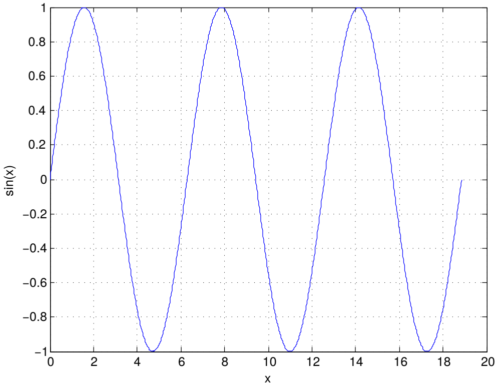
Egenverdier og egenvektorer finner vi også lett

```
A = [2 3 1; 0 -1 4; 2 2 0]
[u,v] = eig(A)
```

$$u = \begin{pmatrix} 0.7470 & 0.7071 & 0.5067 \\ 0.3997 & -0.7071 & -0.8189 \\ 0.5313 & 0.0000 & 0.2695 \end{pmatrix}.$$

$$v = \begin{pmatrix} 4.3166 & 0 & 0 \\ 0 & -1.0000 & 0 \\ 0 & 0 & -2.3166 \end{pmatrix}.$$

Plotting



Plotter to arrays mot hverandre i et koordinatsystem

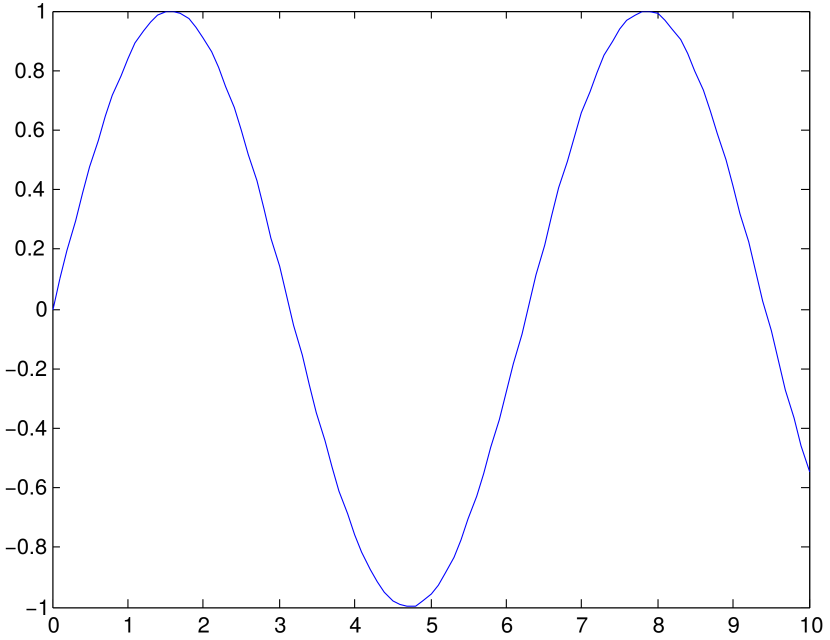
```
x = linspace(0,10,101)
```

$$0, 0.1, 0.2, \ldots, 9.9, 10.$$

```
y = sin(x)
```

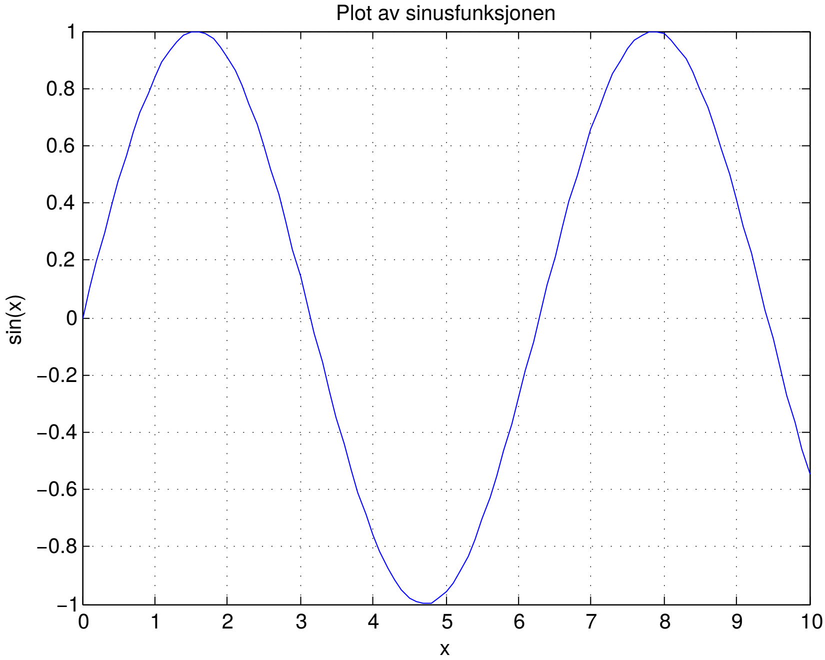
$$\sin(0), \sin(0.1), \sin(0.2), \ldots, \sin(9.9), \sin(10).$$

```
plot(x,y)
```


1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
1
2
3
4
5
6
7
8
9
10

Figurer bør alltid pyntes på litt

```
xlabel('x')
ylabel('sin(x)')
title('Plot av sinuskunsjonen')
grid on
```


Plot av sinusfunksjonen
sin(x)
x
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
1
2
3
4
5
6
7
8
9
10

Vi kan plotte flere kurver i samme figur

`hold on` $\Rightarrow$ Nye plots kommer i samme figur.

`hold off` $\Rightarrow$ Nye plots får ny figur.

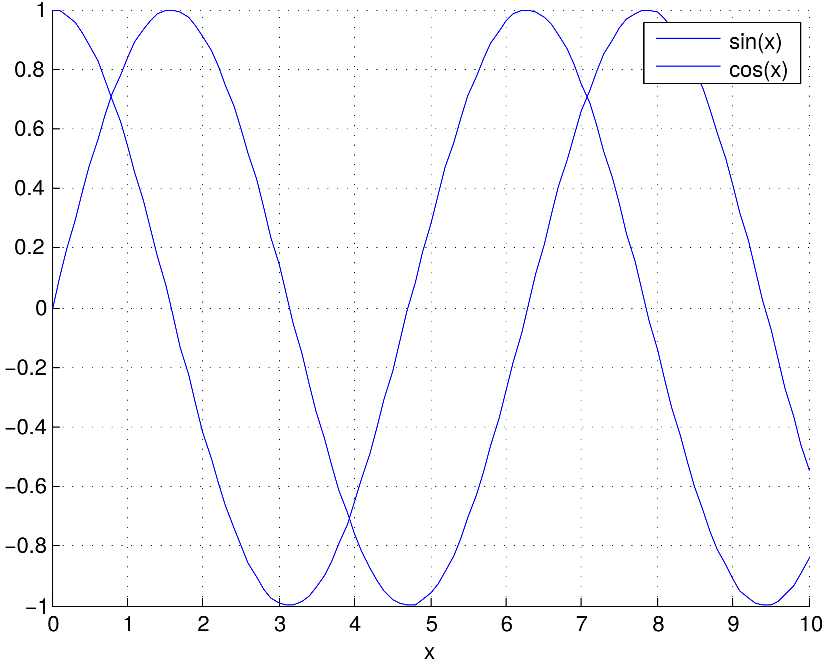
`figure()` $\Rightarrow$ Velger hvilken figur vi jobber med.

```
% Lager arrays med data
x = linspace(0,10,101);
y1 = sin(x);
y2 = cos(x);

% Setter på hold fordi kurvene skal plottes sammen
hold on

% Plotter
plot(x,y1)
plot(x,y2)

% Pynter litt på figuren vi får
xlabel('x')
legend('sin(x)','cos(x)')
grid on
```



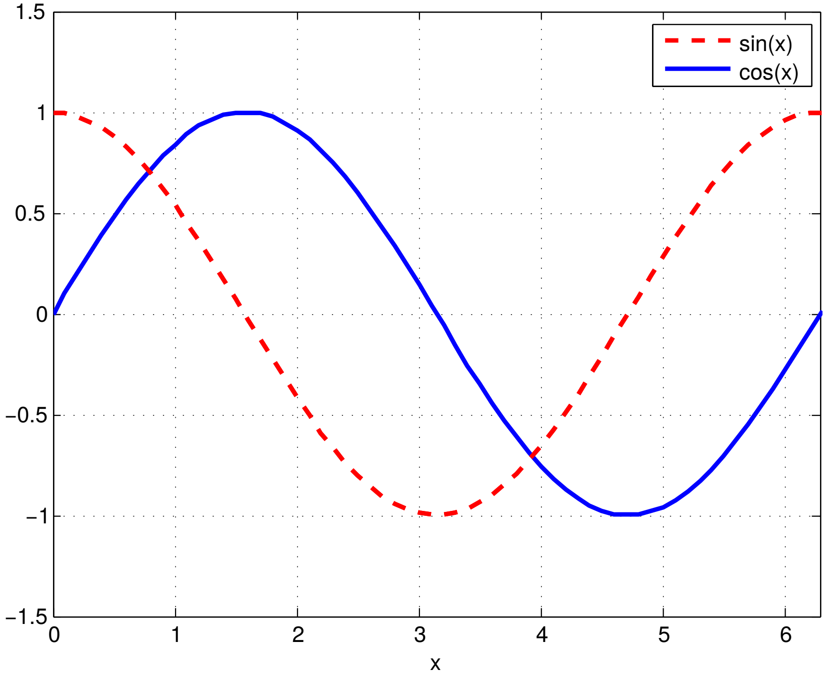
sin(x)
cos(x)
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
0
1
2
3
4
5
6
7
8
9
10
x

Vi kan bestemme utseende på kurvene når vi plotter

```
plot(x, y1, 'b-' ,'linewidth' , 2)
plot(x, y2, 'r--' ,'linewidth' , 2)
```

Bestemmer også hvilke intervaller vi plotter for

```
axis([0, 2*pi, -1.5, 1.5])
```


sin(x)
cos(x)
1.5
1
0.5
0
−0.5
−1
−1.5
0
1
2
3
4
5
6
x

Figurer kan justeres på utrolig mange måter

Det er lurt å se på eksempler.

http://www.mathworks.se/discovery/gallery.html

Programmering i Matlab

Vi programmerer i Matlab ved å skrive scripts

Et script er en tekstfil med Matlab kommandoer.

Lagres som `.m`-filer. Kan da kjøre scripts fra command window.

Vi programmerer i Matlab ved å skrive scripts

Et script er en tekstfil med Matlab kommandoer.

Lagres som `.m`-filer. Kan da kjøre scripts fra command window.

Vi programmerer i Matlab ved å skrive scripts

Et script er en tekstfil med Matlab kommandoer.

Lagres som `.m`-filer. Kan da kjøre scripts fra command window.

Pass på at scrits lagres riktig sted, Matlab kan ikke kjøre scripts den ikke finner.

Funksjoner er spesielle scripts som tar input og gir output

Første linje i en funksjon må være

```
function [y1,...,yN] = myfunction(x1,...,xM)
```

Her er x-ene input og y-ene output.

Funksjoner er spesielle scripts som tar input og gir output

I filen sumAndDiff.m

```
function [s, d] = sumAndDiff(a, b)
% Returns the sum and the difference of two numbers
s = a + b;
d = a - b;
```

I command window:

```
[s, d] = sumAndDiff(5,4)
```

$$\Rightarrow \quad s = 9.$$
$$\Rightarrow \quad d = 1.$$

Vi får ofte bruk for løkker

```
for i=1:10
   disp(i^2)
end
```

$\Rightarrow \quad 1, 4, 9, 16, 25, 36, 49, 64, 81, 100.$

```
s = 1; n = 0;
while (s > 0)
   s = s/2;
   n = n1;
end
disp(n)
```

$\Rightarrow \quad 1075.$

Løkke over en matrise

```
% Find the matrix dimensions
[nx, ny] = size(A);

% Iterate over all matrix elements
for i=1:nx
    for j=1:ny
        % Do something with the matrix element A(i,j)
    end
end
```

If-tester er også viktige

```
% Let x be random number in the interval [0,1)
x = rand(1);

if (x < 0.6)
    disp( 'You loose.' )
else if (0.6 <= x) & (x < 0.99)
    disp( 'You Win!' )
else
    disp( 'You win the Jackpot!' )
end
```

```
% Generate a matrix with random elements
nx = 6; ny = 4;
A = rand(6,4);

% Remove every element smaller than 0.5
for i=1:nx
    for j=1:ny
        if (A(i,j) < 0.5)
            A(i,j) = 0;
        end
    end
end
```

$$\begin{pmatrix} 0.7447 & 0.7802 & 0 & 0.6443 \\ 0 & 0 & 0 & 0 \\ 0.6868 & 0.9294 & 0.5085 & 0.8116 \\ 0 & 0.7757 & 0.5108 & 0.5328 \\ 0 & 0 & 0.8176 & 0 \\ 0.6256 & 0 & 0.7948 & 0.9390 \end{pmatrix}$$

Eksempel

Oppgave:

Lag en funksjon som tar n som input. Funksjonen skal sette opp en $n \times n$ matrise, A, som er gitt ved matriseelementene

$$A_{ij} = \sin\left(\frac{i\pi}{n}\right) + \cos\left(\frac{j\pi}{n}\right).$$

Eksempel

Oppgave:
Lag en funksjon som tar n som input. Funksjonen skal sette opp en $n \times n$ matrise, A, som er gitt ved matriseelementene

$$A_{ij} = \sin\left(\frac{i\pi}{n}\right) + \cos\left(\frac{j\pi}{n}\right).$$

Løsning:

```
function A = buildTrigMatrix(n)
for i=1:n
    for j=1:n
        A(i,j) = sin(i*pi/n) + cos(j*pi/n)
    end
end
```