

Introduktion till MATLAB, med utgångspunkt från Ada

MATLAB

Desktop-miljö som innefattar editor, *kommandofönster*, graffönster och mycket mer.

Interpreteras

- Snabbt att testa kommandon

Särskilt bra för:

- Utföra beräkningar (framför allt med matriser)
- Visualisera data

Ada

Terminal + emacs + gnatmake

Kompileras

- Regjäla kontroller *innan* körning

Särskilt bra för:

- Säkra program
- Realtidssystem

Syntax

MATLAB

Kod sparas i .m-filer

Inget kodskelett behövs.

Satser körs som om de skrivits in i kommandofönstret.
Kallas för *Skript*.

Varje sats står på egen rad.

; undertrycker automatisk utskrift av satsers resultat.

Ada

Kod sparas i .adb-filer

procedure ... is

begin

end ...;

Satser avskiljs med ;

Syntax – Variabler

MATLAB

*behövs ej
variabler skapas då de
tilldelas.*

```
I = 7  
S = 'Maria'  
F = 3.14
```

"Mjuk" typning, d.v.s.

```
F = I
```

Helt ok. F är nu 7

Ada

```
I : Integer;  
S : String(1..5)  
(deklarationer)
```

```
I := 7;  
S := "Maria";  
F := 3.14;
```

Hård typning

Syntax – IO

MATLAB

```
disp(I)  
disp('Hejsan')
```

```
I = input('Mata in ett tal: ')
```

```
fprintf('%4d', I)  
fprintf('%6.3f', F)
```

```
fprintf('%03d', I)  
skriver ut 007
```

Ada

```
Put(I);  
Put_Line("Hejsan!");
```

```
Put("Mata in ett tal: ");  
Get(I);
```

```
Put(I, Width => 4);  
Put(F, fore => 2, aft => 3,  
    exp => 0);
```

Syntax – if-satser

MATLAB

```
if I == 7
```

```
...
```

```
end
```

```
if I > 7 || I == 14
```

```
...
```

```
elseif I ~= 7 && I ~= 8
```

```
...
```

```
else
```

```
...
```

```
end
```

Ada

```
if I = 7 then
```

```
...
```

```
end if;
```

```
if I > 7 or I = 14 then
```

```
...
```

```
elsif I /= 7 and I /= 8 then
```

```
...
```

```
else
```

```
...
```

```
end if;
```

Syntax – loopar

MATLAB

```
for I = 1:10
```

```
...
```

```
end
```

```
while I < 10
```

```
...
```

```
end
```

```
finns ej
```

```
break
```

```
finns ej
```

Ada

```
for I in 1..10 loop
```

```
...
```

```
end loop;
```

```
while I < 10 loop
```

```
...
```

```
end loop;
```

```
loop
```

```
...
```

```
end loop;
```

```
exit
```

```
exit when I = 7;
```

Syntax – Underprogram (1)

MATLAB

finns ej procedurer, out-param.

```
function Y=UP(X)
```

...

```
end
```

Det som finns i returvariabeln (här Y) skickas tillbaka.

Satsen return terminerar funktionen.

Ada

procedure UP(X : out Integer) is...

```
function UP(X : in Integer)
```

```
return Float is
```

```
Y : Float;
```

```
begin
```

```
...
```

```
return Y;
```

```
end UP;
```

Satsen return skickar tillbaka resultatet.

Syntax – Underprogram (2)

MATLAB

```
function Y=UP(X, Z, W)
...
end
```

*flera in-parametrar
(Obs: inga datatyper)*

```
function Y=UP
...
end
```

*inga parametrar, anropas:
var = UP*

```
function UP
...
end
```

*ingen returvariabel, anropas:
UP*

```
function [Y X]=UP
...
end
```

*fler returvariabler, anropas:
[var1 var2] = UP*

Syntax – Underprogram (3)

MATLAB

Ja, på separata .m-filer.
Samma namn som funktionen.

Ja, men kräver att
huvudprogrammet också är en
funktion. Lägggs efter huvud-
funktionen.

Nej

Ada

Underprogram på separata .adb-
filer. Samma namn som
underprogrammet.

Underprogram i samma fil som
huvudprogrammet.

Underprogram i paket.

Syntax – Övrigt

MATLAB

% markerar en kommentar

Skiljer på gemener och
VERSALER.

Ett större kodexempel finns att
titta på på hemsidan
(momstabellen i MATLAB).

Testa help, t.ex.
help disp

Ada

-- markerar en kommentar.

Skiljer **ej** på gemener och
VERSALER.

Syntax – Fält (Vektorer)

MATLAB

A = [1 2 3 4 5] *eller*
A = [1, 2, 3, 4, 5] *eller*
A = 1:5

A(1) = 99

disp(A(1))

A(6) = 0 *vektorn utökas*

disp(A) *funkar fint!*

Ada

type My_Array_Type is
 array (1..5) of Integer;

A : My_Array_Type := (1, 2, 3, 4, 5);

A(1) := 99;

Put(A(1));

A(6) := 0; → **Krasch**

Put(A); → **Kompileringsfel**

Syntax – Vektorer (2)

MATLAB

$A = [1 ; 2 ; 3 ; 4 ; 5]$ *eller*

Skapar en kolumnvektor

$A = [1$
2
3
4
5]

$A = 1:5$

Skapar vektorn [1 2 3 4 5]

$A = 1:2:10$

Skapar vektorn [1 3 5 7 9]

$A = 5:-1:1$

Skapar vektorn [5 4 3 2 1]

Syntax – Vektorer (3)

MATLAB

$A = []$

Skapar en tom vektor

$A = [B \ C]$

Slår ihop vektorerna B och C.

$A = [B ; C]$

Slår ihop vektorerna B och C, vertikalt.

En skalär (ett enkelt tal) anses vara en vektor av längd 1!

Syntax – Matriser

MATLAB

Vektorer är egentligen bara ett specialfall av matriser (flerdimensionella fält).

$M = [1, 2, 3; 4, 5, 6; 7, 8, 9]$

eller

Skapar matrisen

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

eller

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

$M = [1:3 ; 4:6 ; 7:9]$

$M = \text{zeros}(3, 4)$

$M = \text{ones}(3, 4)$

*Skapar en 3x4 matris
med endast nollor
resp. ettor.*

Syntax – Matriser (2)

`M = eye(3)`

Skapar 3x3 enhetsmatris:

1 0 0

0 1 0

0 0 1

`M = diag([1 2 3])`

Skapar diagonalmatris:

1 0 0

0 2 0

0 0 3

`M2 = rot90(M)`

Roterar 90 grader medurs

0 0 1

0 2 0

3 0 0

`M3 = M2'`

Transponerar

0 0 3

0 2 0

1 0 0

Syntax – Matriser (3)

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

$M(r, k)$

Indexering, tar ut element på rad r och kolumn k .

$M(2:3, 2:3)$

Tar ut delmatrisen: rad 2 till 3 och kolumn 2 till 3:
5 6
8 9

$M(\text{end}, \text{end})$

Tar ut elementet på sista raden och sista kolumnen:
9

$M(1, 1:\text{end})$

Tar ut hela första raden:
1 2 3

Syntax – Matriser (4)

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

$L = \text{length}(M)$

Ger största dimensionen, d.v.s. 3. Användbart för vektorer.

$N = \text{numel}(M)$

Ger antal element, d.v.s. 6.

$S = \text{size}(M)$

Ger båda dimensionerna, d.v.s. S blir $[2 \ 3]$.

$R = \text{size}(M, 1)$

$C = \text{size}(M, 2)$

Ger antalet rader resp. antalet kolumner. R blir 2
 C blir 3.

Syntax – Matriser (5)

$v1 =$
1 2 3

$v2 = v1 * 2$

v2 blir [2 4 6]

$v2 = v1 * v1$

Fel!

$v2 = v1 * v1'$

v2 blir 14

$v2 = v1 .* v1$

v2 blir [1 4 9]

$v2 = v1' * v1$

v2 blir

1 2 3

2 4 6

3 6 9

$v2 = v1 ./ v1$

v2 blir [1 1 1]

$v2 = v1 / 2$

v2 blir [0.5 1 1.5]

$v2 = \sin(v1)$

v2 blir [0.8415 0.9093 0.1411]

Syntax – Filer (Text)

MATLAB

```
F = fopen('A.TXT', 'w')  
fprintf(F, '1 2 3\n4 5 6\n')  
fclose(F)
```

```
M = importdata('A.TXT')
```

M blir matrisen

1 2 3

4 5 6

*Finns flera sätt. Läs mer på
sidorna länkade från
help fopen*

Ada

```
Open(F, Out_File, "A.TXT");  
Put_Line(F, "1 2 3");  
Put_Line(F, "4 5 6");  
Close(F);
```

Läsning från textfil

Syntax – Filer (Binär)

$M =$
1 2 3
4 5 6

MATLAB

```
save('A.mat', 'M')
```

```
load('A.mat', 'M')
```

Ada

```
Open(F, Out_File, "A.BIN");  
Write(F, M);  
Close(F);
```

```
Open(F, In_File, "A.BIN");  
Read(F, M);  
Close(F);
```