

Föreläsning 16(Matlab)

DD1315

Programmeringsteknik

7,5 hp

Matlab

- Ett kraftfullt program för att göra beräkningar inom flertalet naturvetenskapliga & tekniska vetenskapsområden.
- Förkortning för MATrix LABoratory
 - Variabler är som standard matriser (rutnät/tabell)
 - Matriser (storlek $n \times m$, $n > 1$, $m > 1$)
 - Vektorer (en matris med en rad eller en kolumn)
 - Skalär (en matris med en rad och en kolumn)

Generellt

- En matlabfil har ändelsen ".m"
- För att undertrycka utskrift skriver man ett semikolon (;) i slutet av godtycklig sats.
- Operationer mellan variabler följer som standard vektoralgebraiska regler, vill man att beräkningar ska ske elementvis får man skriva en punkt (.) framför operatören.
- Kommentarer inleds med procent (%)
- Kommandot `help <kommando>` är ofta användbart.

Variabler

- Hakparenteser används för att definiera matriser. Inom dessa används mellanslag för att skilja mellan kolumner och semikolon för att skilja mellan rader.
- `temperatur = [12.1 9.6 13.5]; % Dimension 1x3`
- `A = [1 2 ; 3 4]; % Dimension 2x2`
- `size(variabelnamn)` ger en variabels dimension

In & utmatning

- Inmatning sker med funktionen *input()* som returnerar inmatningen i form av lämplig datatyp, d v s kan indata tolkas som ett tal så blir det ett tal, ingen typkonvertering behövs.
- Utmatning sker med funktionen *disp()* som kräver en matris (vektor) med strängar som argument, värden som är tal behöver därför typkonverteras innan utmatning med funktionen *num2str()*

Funktioner

- Funktioner definieras i en egen fil och anropas genom att ange filens namn (se till att spara alla .m-filer i samma katalog).
- Det är här viktigt att tänka på vad som är skalärt och vektorvärt bland parametrarna.

Exempel

```
function f = cirkelarea(r) % Sparas i cirkelarea.m  
f = pi*r.^2;
```

Exempel

```
% inmatning, funktionsanrop, utmatning
```

```
clear
```

```
r = input('Ange cirkelns radie: ');
```

```
disp(['Cirkelarean =' num2str(cirkelarea(r))])
```


Grafer, 2D

- För en vanlig funktionskurva används vanligtvis funktionen *plot* som tar två stycken vektorer som argument (x,y)
 - hold on (graf på graf)
 - grid (ett rutnät för ökad läsbarhet)
 - xlabel (x-axelns titel)
 - ylabel (y-axelns titel)
 - title (grafens titel)
 - subplot (för indelning av figurfönstret i delar)
 - clf (för att rensa figurfönstret)

Exempel

```
function f = graf2d(x) % sparas i graf2d.m  
f = sin(x)./x;
```

Exempel

```
% exempel2d.m  
clear, clf  
x = -4*pi:0.5:4*pi;  
y = graf2d(x);  
plot(x,y)  
grid  
xlabel('x')  
ylabel('y(x)')  
title('Graf')
```

Grafer, 3D

- Först anropas *meshgrid()* som returnerar två stycken matriser innehållandes
 - x-värdena för hela domänen.
 - y-värdena för hela domänen
 - Med domän avses här den yta i x,y-planet som det ska beräknas ett z-värde för.
- Sedan anropar man *mesh(x,y,z)* för att rita.

Exempel

```
function f = graf3d(x,y) % sparas i graf3d.m  
f = sin(sqrt(x.^2 + y.^2))./(sqrt(x.^2+y.^2));
```

Exempel

```
% exempel3d.m
```

```
clear
```

```
clf
```

```
[x,y]= meshgrid(-4*pi:0.5:4*pi,-4*pi:0.5:4*pi);
```

```
z=graf3d(x,y);
```

```
mesh(z)
```