

Ok, jeg har aldri skrevet et krasjkurs i programmering før, send meg epost om det er noe du ønsker skal være med her. Jeg kommer til å holde det helt enkelt.

Komme igang

1 Installer Matlab: <https://innsida.ntnu.no/wiki/-/wiki/English/Matlab+for+students>

2 Åpne Matlab

3 Gå til kommandovinduet og skriv

`demo`

4 Trykk på ENTER

Når du trykker på ENTER, blir du tatt videre til et sted der du kan få vite alt du lurer på. Hvis du ikke liker internett eller engelsk så godt, kan du fortsette å lese her.

Bruksanvisningen for dette krasjkurs er veldig enkel: Alt som står i en slik ramme:

`demo`

skal du skrive inn i kommandolinjen og trykke på ENTER.

Dersom det er bare en linje, kan du bare skrive det rett i kommandovinduet. Dersom det er flere linjer, bør du klippe det ut og lime det inn i en fil, som du for eksempel kaller 'test.m', eller noe i den dur. Hvis du skriver

`test`

i kommandovinduet, vil alle linjene i filen kjøres. NB: husk å stå i rett katalog i matlab (den katalogen der test.m er lagret), og husk å lagre filen før du kjører.

Grunnleggende regneoperasjoner

MATLAB står for MATriseLABoratorium, og er ikke mye vanskeligere å bruke enn en kalkulator. Det er glitrende for matriseregning. Jeg bruker det som kalkulator, når du blir vant til det går jo alt mye kjappere.

Vi tar de vanlige regneoperasjonene først. Addisjon:

`2 + 3`

Subtraksjon

`2 - 3`

Multiplikasjon

`2 * 3`

Divisjon

`2 / 3`

Divisjon er litt vittig, for den kan brukes begge veier. Backslash får vi bruk for siden, så det er lurt å finne den på tastaturet ditt.

`2 \ 3`

Potenser går fint

`2^3`

Kvadratrotter tar du med potens

`print 2^.5`

eller

`sqrt(2)`

Husk å alltid bruke gangetegn!

`2(2 + 3)`

`2*(2 + 3)`

Hvis du ikke ønsker utskrift av det du gjør, skriver du bare

`2*(2 + 3);`

Variable

Vi lagrer variable slik:

`a=2*(2 + 3);`

`b=7;`

Skriver du

`a`

får du en utskrift av verdien lagret i *a*. Du kan herje med variable akkurat som med tall

`a+b`

Hvis du er lei av verdien $a = 12$, kan du overskrive den

`a=2;`

eller skrive

`clear a`

Kommandoen

`clear`

sletter alle variable.

Vektorer

Matlab behandler vektorer og matriser akkurat på samme måte som vi gjorde i M3. Vi begynner med å lage en rekkevektor

```
x=[2 5 7]
```

eller en søylevektor

```
y=[3; 6; 8]
```

Hvis du har en rekkevektor, og skulle ønske at det var en søylevektor, skriver du

```
x'
```

Du kan også skrive

```
y=[2 5 7]'
```

for å lage søylevektor. Hvis du ønsker å plukke ut komponenter i en vektor, skriver du

```
y(2)
```

Matlab følger vanlige regneregler for matriser og vektorer.

```
x+y
```

Søylevektorer kan legges sammen

```
x'+y
```

og rekkevektorer også

```
x'+y
```

Vi kan også gange vektorer med skalarer

```
2*x
```

og ta indreprodukt

```
x*y
```

og ytreprodukt

```
y*x
```

Plotting

Nå kan vi nok om Matlab til å plotte funksjoner. Her er en ting som er viktig å forstå: Matlab plottet vektorer mot hverandre, så man lager først en vektor med verdier på x -aksen, og så en vektor med funksjonsverdier. Matlab tegner en rett strek mellom punktene, men hvis det er tett nok mellom verdiene på x -aksen vil kurven se glatt ut.

La oss plotte funksjonen $f(x) = \sin x$ på intervallet $[0, 2\pi]$. Først må vi lage en vektor med punkter på x -aksen

```
x=0:.01:2*pi
```

Denne kommandoen lager et gitter med verdier fra 0 til 2π . Gitterfinheten er 0.01. Så lager vi en vektor med funksjonsverdier

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

Nå skriver vi

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

tegnes funksjon i et nytt vindu. For å få en illustrasjon av hvordan Matlab tegner funksjoner kan du plotte på nytt

```
x=0:.5:2*pi;  
y=sin(x)  
plot(x,y)
```

Stikkordet i Matlab er altså å tenke vektorer istedet for funksjoner, og det er noen regneregler må man vite litt om. Spesielt gangning og potenser er litt annerledes. Vi illustrerer med å plotte noen flere funksjoner. Lag et nytt intervall på x -aksen

```
x=0:.01:1;
```

og en vektor med funksjonsverdier

```
y=x.*sin(x);
```

Denne kommandoen illustrerer to viktige ting. Gangetegnet $*$ er reservert for matriseprodukter, så hvis man ønsker å gange vektorer komponentvis til en ny vektor (en operasjon vi sjelden eller aldri bruker i matematikk) må du skrive gangetegn med punktum foran. Sinusfunksjonen har Matlab laget slik at den skjønner av seg selv at den skal lage en vektor med sinusfunksjonen evaluert på alle punktene i vektoren x , så det går av seg selv. Potenser oppfører seg likt som gangning (husk punktum)

```
z=x.^2.*sin(x);
```

Hvis du har lyst til å plotte to funksjoner i samme figur, skriver du

```
figure  
hold on  
plot(x,y)  
plot(x,z)
```

Du kan bestemme farge på grafene i....

Forresten en helt annen ting. Hvis du vil vite hva matlab sier om sin egen plottefunksjon, skriver du

```
help plot
```

Da får du info om hvordan plot funker. Og der vil du se at du kan bestemme farge på grafene

```
figure  
hold on  
plot(x,y,'g')  
plot(x,z,'b')
```

eller gjøre andre gøyete ting. En liste over alt du kan gjøre, får du når du skriver help plot.

For og if

En for løkke er en måte å få datamaskinen til å gjøre samme ting om igjen, men med forskjellige parameter

```
for i =1:5
    disp(i)
end
```

og if kan brukes til å gjøre unntak

```
for i =1:5
    if i==3
        disp(i)
    end
end
```

Forresten har indentering ikke noe å si i Matlab, ut-
over å gjøre koden mer lesbar.

Matriser

3D-plot

Animasjon