

---

# Matematikk 1000

## Øvingsoppgaver i numerikk – leksjon 3

### *Funksjoner og plotting*

---

I denne øvinga skal vi først og fremst lære oss å lage plott i MATLAB.

Ellers minner vi om at der er mange MATLAB-ressurser tilgjengelig. Noe skriftlig materiale er tilgjengelig på Fronter-sidene. Man kan søke etter dokumentasjon i MATLAB selv, eller man kan “google” etter hjelp. Sist, men ikke minst, man kan spørre lærerne om hjelp.

Det forutsettes at man har gjort leksjon 1.

I en del av disse oppgavene er det mye tekst. Det betyr ikke nødvendigvis at det krever mye arbeid å gjøre dem. Om det oppleves som mye arbeid: Spør om hjelp!

Vi har sett at en hel del funksjoner allerede er lagt inn i MATLAB. Dette gjelder for eksempel de trigonometriske funksjonene – med inversfunksjoner. De heter `sin`, `cos`, `tan`, `asin`, `acos` og `atan`. Eksponentialfunksjonen,  $e^x$ , og den naturlige logaritme-funksjonen,  $\ln x$ , finnes også; de heter `exp` og `log`. I tillegg kan vi lage våre egne funksjoner. Det skal vi se mer på i neste leksjon.

### Oppgave 1 – Å lage et plott

- La vektoren  $\mathbf{x}$  starte med 0 og så gå i steg på 0.1 opp til  $7\pi$ . (Med fare for å være over-tydelig: Denne tilordninga skal utføres i kommandovinduet i MATLAB.)
- La vektoren  $\mathbf{y}$  bestå, elementvis, av sinusverdiene av vektoren  $\mathbf{x}$ . Det gjør du slik i MATLAB: `'> y=sin(x);'`. Denne funksjonen kan ta både skalare (tall-) argumenter og vektorer, slik som her.
- Plott sinusfunksjonen: `'> plot(x,y)'`.
- Undersøk hva som skjer når du skriver `'plot(x,y,'r)'`, `'plot(x,y,'linewidth',3)'` og `'plot(x,y,'g- -)'`.
- Bruk MATLAB til å plote følgende funksjoner:

- i)  $f(x) = \sqrt{x} + \ln x$  med definisjonsmengden<sup>1</sup>  $D_f = [1/4, 5]$ .
- ii)  $g(x) = \arctan x$  med definisjonsmengden  $D_g = [-10, 10]$ .  
("arctan  $x$ " er det samme som " $\tan^{-1} x$ ".)
- iii)  $h(x) = \cos(2x) \cdot e^{-x^2/10}$ . Velg selv et passende intervall for  $x$  her.

I alle tre tilfellene: Pass på at steglengda i  $x$ -vektoren din er liten nok til at grafen ikke blir hakkete.

## Oppgave 2 – Å lagre et plott

Når vi skal lagre noe, er det selvsagt viktig at vi har kontroll på hvor det blir lagra. Det er nok en god idé å opprette ei mappe for de MATLAB-relaterte filene som etterhvert blir oppretta i dette kurset. Det vil nok bli nødvendig å dele denne inn i flere under-mapper også.

Det aller meste av det som har med lagring å gjøre, kan gjøres ved hjelp av klikkbare menyer. Men det kan også gjøres med kommandoer i kommandovinduet. Selv om de fleste antakeligvis vil foretrekke "klikking", vil vi kort forklare hvordan ting gjøres fra kommandovinduet også.

- a) Når du starter MATLAB, vil du i utgangspunktet bli plassert et eller annet sted i fil-systemet ditt. Hvor dette er, ser du i mappevinduet. (Alternativt kan du i kommandovinduet finne ut hvilken mappe du er i ved å skrive `pwd` – *print working directory*.)

Ta utgangspunkt i ett av plottene du lagde i oppgave 1). Dette kan du lagre ved å klikke **Save** i **File**-menyen oppe til venstre i vinduet med plottet. Gjør dette og pass på at figuren blir lagra i ei passende mappe. Figuren bli da lagra som ei **fig**-fil – MATLABs eget figurformat. Prøv gjerne å lukke figuren og åpne den igjen.

- b) Man kan fint skrive ut grafer direkte fra MATLAB. Men ofte kan det være en fordel å lagre figuren som ei bildefil av et eller annet slag først. Dette er nødvendig å gjøre om du ønsker å ta med plott i andre typer dokumenter – som for eksempel tekstbehandlingsfiler (OpenOffice, Word etc.). De vanligste tilgjengelige formatene er **jpg/jpeg**, **png** og **pdf**<sup>2</sup>.

Når man skal lagre et plot som ei bildefil, kan man også gjøre dette ved å bruke **Save** eller **Save As** i fil-menyen oppe til venstre i figur-vinduet. Du velger det formatet du vil ha i **Save as type**-menyen nederst i det vunduet som dukker opp – og passer på å plassere fila i den katalogen du vil ha den i før du trykker **Save**.

Alternativt kan det hele gjøres fra kommandolinja:

<sup>1</sup> *Definisjonsmengden* til en funksjon er mengden av de tillatte  $x$ -verdiene.

<sup>2</sup> Forkortelsene står for henholdsvis *Joint Photographic Expert Group*, som kom opp med **jpeg**-formatet, *Portable Network Graphics* og *Portable Document Format*.

```
>> print -f<figurnummer> -d<format> <Filnavn>.<format>}
```

Om du for eksempel vil at figur 2 – man kan ha flere figurer oppe samtidig – skal eksporteres til `png`-formatet og hete `FigurenMin` (i tillegg til “etternavnet” `png`), skriver du

```
>> print -f2 -dpng FigurenMin.png
```

Når du gjør dette, blir figuren lagra i den arbeidsmappe du er i; om du vil lagre den et annet sted, kan du manøvrere deg dit<sup>3</sup>.

Velg en av metodene over og lagre plottet ditt som ei `png`-fil i ei passende mappe. Forsøk å åpne `png`-fila med en eller annen bilde-applikasjon eller å legge det inn i et tekstbehandlingsprogram.

### Oppgave 3 – Flere grafer samtidig

- Lag et plott av grafen til funksjonen  $f(x) = x^2 + 1$ . Velg selv hvilket intervall  $x$  skal tilhøre og hvor fin inndeling du vil ha på den tilsvarende vektoren. (Tips: Gjør som over. Lag først en vektor med de aktuelle  $x$ -verdiene ved å bruke skrivemåten `a:d:b` – du velger selv start og slutt, `a` og `b`, og *steglengda* `d`. Så bruker du denne  $x$ -vektoren til å lage en  $y$ -vektor som du kan bruke i `plot`-kommandoen. Pass på at `d` blir så liten at figuren ikke ser hakkete ut.)
- Lag en vektor med funksjonsverdier for funksjonen  $g(x) = e^x$  med den samme  $x$ -vektoren som i a). Lag et plott av  $g(x)$  også. Lag en figur som inneholder både grafen til  $f$  og  $g$  samtidig. Dette kan gjøres på to måter. Enten slik:

```
>> plot(x,y,x,z)
```

eller slik:

```
>> plot(x,y)
>> hold on
>> plot(x,z)
>> hold off
```

Her har vi valgt å kalle vektorene med funksjonsverdier for  $f$  og  $g$  henholdsvis `y` og `z`.

- Vi kan “zoome” inn på en mindre del av grafen. Dette kan gjøres med musa om vi bruker funksjonen markert med et forstørrelsessglass i plottvinduet over selve plottet. Man kan også gjøre det på kommandolinja ved hjelp av funksjonen `axis`. Om vi for eksempel skriver

---

<sup>3</sup>Man kan som sagt manøvrere mellom ulike mapper i mappevinduet. Men også dette kan gjøres i kommandovinduet. Man bruker å så fall kommandoen `cd`, *change directory*, på samme måte som i `dos`-vinduet i Windows eller i terminalvinduet i Linux/Unix.

```
>> axis([-2 3 -1 4])
```

vil  $x$ -aksen gå fra -2 til 3, og  $y$ -aksen vil gå fra -1 til 4. Ved hjelp av denne funksjonen, avgrens plottet til  $x$ -verdier du velger selv og finn ei passende avgrensing for  $y$ -aksen til.

## Oppgave 4 – Litt cosinus-leik

For  $x$  i intervallet  $[0, 2]$ , plott grafene til følgende tre funksjoner samtidig:

- $a(x) = \cos(2\pi x)$
- $b(x) = \cos(\pi x)$
- $c(x) = 2 \cos(2\pi x)$
- $d(x) = \cos(\pi x^2)$

La grafen til  $a$  være svart ('k') og heltrukket, la grafen til  $b$  være rød og stipla, grafen til  $c$  skal være blå og "prikka" ('b:') og  $d$  skal få en "punkt-strek"-graf som er grønn ('g-.').

Til sist: Lag en liten firkant som forklarer hvilken graf som er hva:

```
>> legend('a(x)', 'b(x)', ...)
```

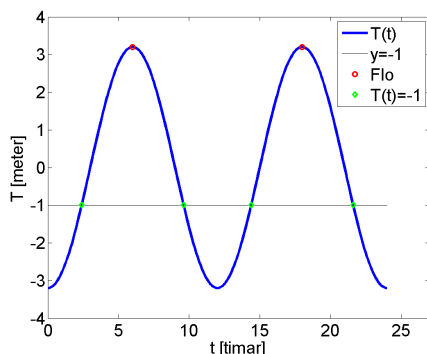
## Oppgave 5 – Flo og fjære

Figur 1 er laget slik:

```
>> x=0:.1:24;
>> y=3.2*sin(pi/6*(x-3));
>> plot(x,y,'linewidth',3)
>> set(gca,'fontsize',20)
>> xlabel('t [timar]'); ylabel('T [meter]')
>> hold on
>> plot([0 24],[-1 -1],'k-')
>> plot([6 18],[3.2 3.2],'ro','linewidth',2)
>> plot([2.39 9.61 14.39 21.61],[-ones(1,4),'gd','linewidth',2)
>> legend('T(t)', 'y=-1', 'Flo', 'T(t)=-1')
>> axis([0 27 -4 4])
>> hold off
```

Plottet viser en enkel modell for hvordan vannstanden varierer gjennom døgnet et bestemt sted. I tillegg er det markert når det er flo, og når vannstanden er  $-1$  m.

- a) Utfør disse kommandoene en etter en og forsøk å forstå hva hver enkelt av dem gjør. ('gca' står for 'get current axis', og 'ones(1,4)' lager en vektor med 4 elementer som alle er 1.)



Figur 1: Figuren viser en modell for flo og fjære. De røde sirklene markerer flo, og de grønne “diamantene” markerer tidspunktene da vannstanden var 1 m under normalen.

- b) Etter ha skrevet inn ei slik remse med kommandoer, ville det være surt å oppdage at du heller ville skrive “timer”, bokmål, i stedet for “timar”, som det heter på nynorsk, på  $x$ -aksen. Kanskje finner du også ut at du egentlig ville ha en sort graf – ikke en blå. I så fall må jo to kommandoer endres, og dermed må nok også de andre kommandoene repeteres. Å repetere denne remsa går for så vidt nokså raskt om vi bruker piltastene til å bla oppover i kommandohistorikken. Men det blir ganske tungvidt likevel.

Den gode nyheten er at vi kan godt velge å skrive denne remsa som ei tekstfil og så henvise MATLAB til denne. Tekstfila lager vi i en *editor*. Bruk gjerne MATLABs egen editor; den kan du starte ved å trykke på “*New Script*” oppe til venstre. Skriv inn den sekvensen av kommandoer du nettop har utført inn i editoren (her kan du nok spare litt tid ved å kopiere og lime fra kommandovinduet.) Lagre denne fila i ei passende mappe og gi den navnet `FloFjaere.m`. Om du i “Current Folder”-vinduet manøvrerer deg fram til samme mappe som denne fila ligger i, kan man ganske enkelt skrive ‘> FloFjaere’ (uten ‘.m’) i kommando-vinduet, og denne sekvensen med kommandoer blir utført<sup>4</sup>.

Gjør dette.

En slik sekvens av kommandoer kaller vi et *skript*. Slike vil vi definitivt komme tilbake til.

<sup>4</sup>Man kan også kjøre et skript ved å trykke F5-tasten fra editor-vinduet eller ved å klikke på den grønne pila som det står \* ‘run’ under oppe i menylinja i editoren.