
Matematikk 1000

Øvingsoppgaver i numerikk – leksjon 3

Skript

I denne øvinga skal vi lære oss å lage *skript*. Et skript kan vi se på som et lite program – altså en sekvens av kommandoer. Dette er noe vi kommer til å bruke mye i MATLAB.

I leksjon 2 introduserte vi *if*-satser. De skal vi se mer til. Vi kommer også her til å introdusere et nytt begrep: *for*-løkker. I dette settet skal vi blant annet bruke slike til å regne ut lange summer, summer du ikke har lyst til å regne ut på kalkulator eller med papir og blyant.

Det forutsettes at du har gjort de tidligere oppgavesettene.

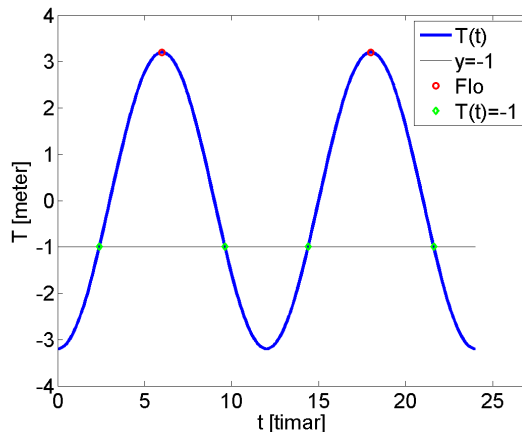
Oppgave 1 – Flo og fjære

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 “timar”, 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*. Som vi så i leksjon 2, har MATLAB sin egen teksteditor, som du kan åpne ved å klikke på “New Script” oppe til venstre. Gjør dette og skriv inn den sekvensen av kommandoer du nettop har utført. (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 `FloFjære.m`. Om du i “Current Folder”-vinduet manøvrerer deg fram til samme mappe som denne fila ligger i, kan man ganske enkelt skrive `» FloFjære` (uten `‘.m’`) i kommando-vinduet, og denne sekvensen med kommandoer blir utført¹.

Gjør dette.

En slik sekvens av kommandoer, kaller vi et *skript*.

¹Man kan også kjøre et skript ved å trykke F5-tasten fra editor-vinduet.

Oppgave 2 – Summer og for-løkker

Det er kjedelig å gjøre den samme jobben, eller nesten den samme jobben, om igjen og om igjen. Derfor er det bra vi har datamaskiner. I denne oppgava introduserer vi **for**-løkker, som er et veldig nyttig verktøy når man skal gjenta mer eller mindre like operasjoner mange ganger.

- a) Skriv ut alle leddene i summen $\sum_{i=1}^{10} i^2$ ned på papir. Hva blir summen?
- b) Åpne teksteditoren og skriv av følgende tekst:

```
sum=0;
for i=1:10
    sum=sum+i^2;
end
sum
```

Lagre fila i ei passende mappe og gi den navnet **FinnSum.m**.

- c) Manøvrér deg til den katalogen hvor fila ligger, og skriv '**» FinnSum**' i kommandovinduet. Får du opp svaret frå oppgave a)?

Dette skriptet inneholder ei **for**-løkke; den starter med '**for i=1:10**' og slutter med '**end**'. Det som skjer når skriptet kjører, er at det som står inni løkka, **sum=sum+i^2**;, blir gjentatt i tur og orden for alle de **i**-verdiene som blir angitt. Første gang er **i** lik 1. Da blir 1^2 lagt til variabelen **sum**, slik at **sum** blir endra fra 0 til 1. Når vi kommer til **end**, går vi tilbake til **for**-linja og setter **i** til å være 2. Så blir 2^2 lagt til **sum**, slik at denne variabelen nå blir 5. Dette gjentar vi til og med at **i** er 10. Når vi har lagt til 10^2 , er vi ferdig med løkka. Da blir den siste kommandoen, **sum**, utført. Siden det ikke står noe semikolon bak, blir verdien av **sum** skrevet til skjerm.

Ta gjerne bort semikolonet i linje 3 for å se hva som skjer fra gang til gang.

Det er vanlig å ha et par innrykk på den delen av skriptet som står inni selve løkka; dette gjør skriptet lettere å lese. Denne "innmaten" kan bestå av flere kommandoer – ikke bare én som her.

- d) Justér skriptet over slik at den regner ut $\sum_{i=1}^{100} i^2$. (Dette hadde tatt ganske lang tid å gjøre for hånd, hadde det ikke?)
- e) Endre skriptet ditt slik at dt regner ut summen $\sum_{i=5}^{20} \sqrt{i}$.
- f) Hvilken sum blir beregna i dette skriptet?

```
sum=0;
for i=0:12
    sum=sum+1/i;
end
sum
```

Gi svaret ved hjelp av Σ -notasjon.

Et viktig poeng til sist: Skriptet over har en alvorlig mangel; det er ikke *kommentert*. Å kommentere et skript vil si at vi legger inn tekst som forklarer hva skriptet gjør. Dette kan vi gjøre ved hjelp av %-tegnet; alt som står bak et prosenttegn, blir ignorert av MATLAB – men ikke nødvendigvis av en person som leser skriptet. En kommentert versjon av skriptet over kan se slik ut:

```
% Dette skriptet beregner summen av i^2 for i fra og  
% med 1 til og med 10.  
  
sum=0;           % Setter summen til 0 til å begynne med  
for i=1:10  
    sum=sum+i^2;  % For hver i: Legger til i^2  
end  
sum              % Skriver summen til skjerm
```

Å legge til slik tekst kan synest ganske unødvendig når man nettopp har skrevet skriptet; “det er jo så opplagt hva det gjør!”. Vi er nok mange som har erfart at slike ting slett ikke er så opplagt når man skal se på skript man har skrevet tidligere – for ikke å snakke om hvis man skal bruke skript som noen andre har skrevet. Du kan for så vidt gjøre deg denne erfaringa selv også, men jeg anbefaler deg å tro på oss når vi sier at det *alltid* er en god idé å kommentere skriptene du skriver.

Oppgave 3 – Lys til bløtkaka

I denne oppgava tenker vi oss at en familie alltid feirer bursdagen til barna med bløtkake med lys på – med like mange lys som barnet fyller år. Oppgavene kan løses med papir og blyant, men de kan også løses ved å skrive eit lite MATLAB-skript. Forsøk gjerne begge deler.



- a) Da et av barna ble født, fikk det en pakke med 10 kake-lys i presang. Lysene er så små at de ikke kan brukes om igjen. Hvor lenge vil denne pakka vare?
- b) Om vi i stedet hadde hatt en pakke med 100 kakelys, hvor lenge ville *den* vart?
- c) En annen familie med samme bursdags-praksis har to barn, Kari på 7 år og Ola på 5. Ved nyttår har foreldrene 300 lys. Når må de kjøpe flere?

Ekstra: Oppgave 4 – Maksimering

Lag ei funksjonfil som tar en vektor $\mathbf{x}$ som argument (*input*) og returnerer den største verdien i $\mathbf{x}$ (*output*)².

²MATLAB har rettnok denne funksjonen fra før; den heter `max`. Men det kan jo være en god øvelse å lage den selv også