

---

# Matematikk 1000

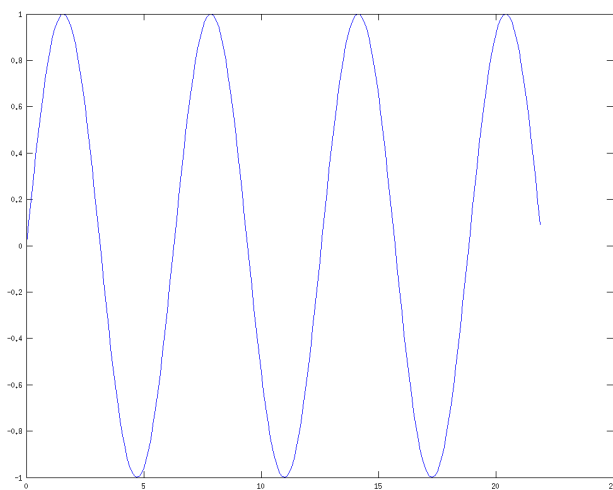
## Øvingsoppgaver i numerikk – leksjon 3

### Løsningsforslag

---

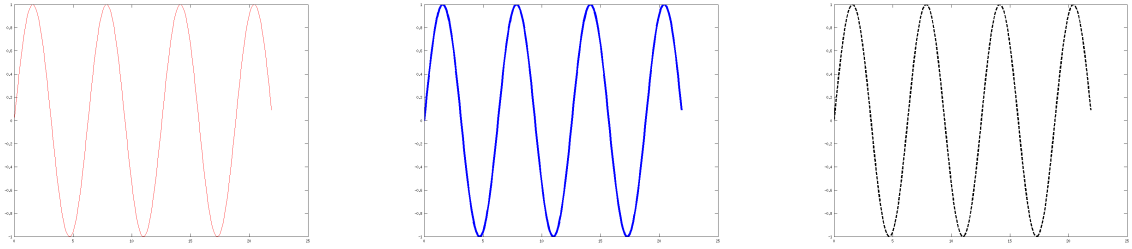
#### Oppgave 1 – Å lage et plott

- a) Vi kan tilordne vektoren slik i kommandovinduet: `'> x=0:.1:7*pi;'`  
Legg merke til at det ikke er opplagt hvordan dette skal tolkes; skal vektoren `0:.1:7` ganges med  $\pi$  eller skal vektoren gå opp til  $7\pi$ ? Det viser seg at MATLAB tolker det som sistnevnte. For å unngå tvil, kunne vi ha skrevet `'> x=0:.1:(7*pi);'`.
- c) Kommandoen `'> plot(x,y)'` gir figur 1.



Figur 1: Plot av funksjonen  $\sin x$ .

- d) `'> plot(x,y,'r)'` gir en rød kurve i stedet for blå, som er “default”, og `'> plot(x,y,'linewidth',3)'` gir en tykkere kurve. Ved å sette en enda høyere verdi enn 3, får man en enda tykkere graf. Ei stipla kurve får man ved å skrive `'--'`; dette gir for eksempel ei nokså tjukk, svart kurve med stipla graf: `'> plot(x,y,'k--','linewidth',2)`. Se figur 2.



Figur 2: Plot av funksjonen  $\sin x$  i ulike former.

e) Funksjonen  $f(x)$  kan plottes på denne måten:

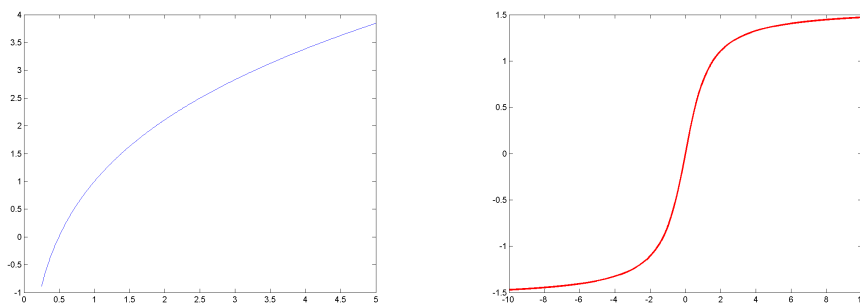
```
>> x=.25:.01:5;
>> f=sqrt(x)+log(x);
>> plot(x,f)
```

Resultatet er vist til venstre i figur 3.

Funksjonen  $g(x)$  kan plottes på tilsvarende måte. Her har vi valgt steglengda 0.001 for  $x$ -vektoren:

```
>> x=-10:1e-3:10;
>> g=atan(x);
>> plot(x,g,'r','linewidth',2)
```

Vi har her valgt at grafen skal være rød og noe tykkere. Resultatet er vist til høyre i figur 3.



Figur 3: Plot av funksjonene  $f(x)$  og  $g(x)$  i oppgave 1 e).

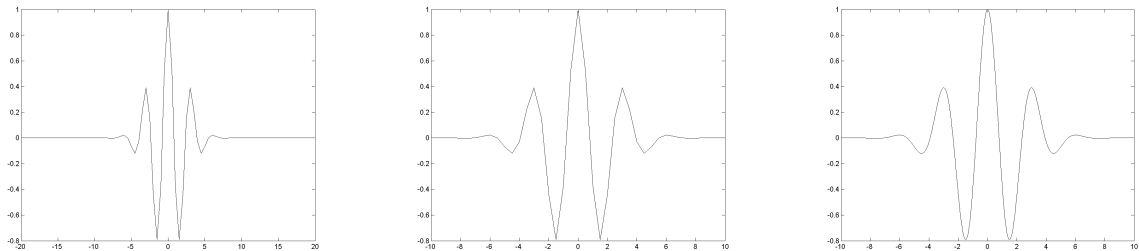
Det er ikke alltid like lett å finne ett passende  $x$ -intervall når man skal lage et plott. Men om vi først plottes  $h(x)$  over et stort  $x$ -intervall, ser vi snart at  $[-10, 10]$  kan være et bra valg. Se venstre del av figur 4. I den midtre delen av av figuren, har vi plotta  $h(x)$  på denne måten:

```
>> x=-10:.5:10;
>> h=cos(2*x).*exp(-x.^2/10);
>> plot(x,h,'k')
```

Som vi ser, ble dette noe hakkete; vi har nok valgt for stor steglengde i  $x$ . Vi gjør det om igjen med 0.001 som steglengde i stedet for 0.5:

```
>> x=-10:1e-3:10;
```

Resultatet er vist til høyre i figur 4.



Figur 4: Ulike plot av funksjonen  $h(x)$  i oppgave 1 e.

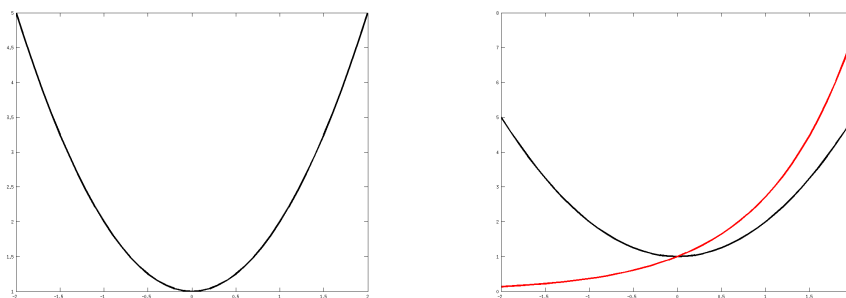
s

### Oppgave 3 – Flere grafer samtidig

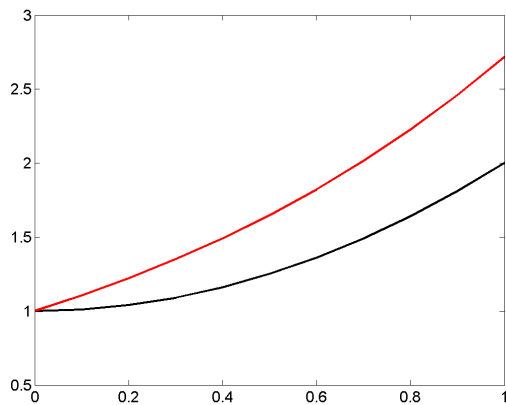
a) Dette kan for eksempel gjøres slik:

```
>> x=-2:.1:2;  
>> y=x.^2+1;  
>> plot(x,y,'k','linewidth',2)
```

Resultatet ser du i til venstre i figur 5.



Figur 5: Plott av funksjonen  $f(x) = x^2 + 1$  (svart) og funksjonen  $g(x) = e^x$  (rød).



Figur 6: Samme som høyre del av figur 5 men med andre grenser for aksene.

b) Med disse kommandoene:

```
>> z=exp(x);
>> hold on
>> plot(x,z,'r','linewidth',2)
>> hold off
```

får vi plottet til høyre i figur 5. Merk at uten `hold on`-kommandoen, ville plottet av  $f(x)$  ha blitt fjerna da vi plotta  $g(x)$ .

c) Kommandoen `axis([0 1 .5 3])` gir plottet i figur 6. Vi har her avgrensa  $x$ -aksen til å gå fra 0 til 1 og  $y$ -aksen til å gå fra 1/2 til 3. Merk at tallene på aksene har en tendens til å bli veldig små. I figur 6 har vi gjort tallene større ved å skrive

```
set(gca,'fontsize',15)
```

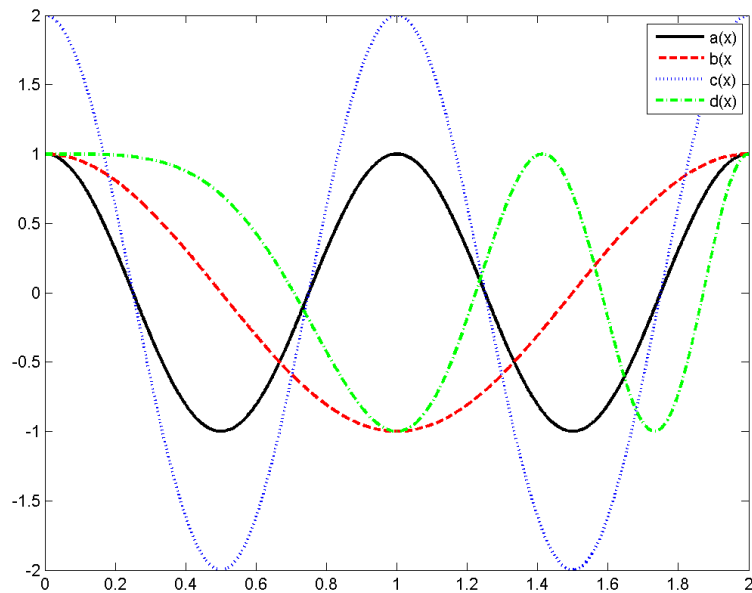
i kommandovinduet<sup>1</sup>.

## Oppgave 4 – Litt cosinus-leik

Vi kan skrive følgende i kommandovinduet i MATLAB:

```
>> x=0:1e-3:2;
>> a=cos(2*pi*x);
>> b=cos(pi*x);
>> c=2*cos(2*pi*x);
```

<sup>1</sup>'gca' står for "get current axis".



Figur 7: Plottet vi skulle lage i oppgave 4.

```
>> d=cos(pi*x.^2);
>> plot(x,a,'k','linewidth',2)
>> hold on
>> plot(x,b,'r--','linewidth',2)
>> plot(x,c,'b:','linewidth',2)
>> plot(x,d,'g-.','linewidth',2)
>> hold off
>> legend('a(x)', 'b(x)', 'c(x)', 'd(x)')
```

Plottet vi får, er vist i figur 7.

## Oppgave 5 – Flo og fjære

```
a) >> x=0:.1:24;
>> y=3.2*sin(pi/6*(x-3));
Disse linjene burde vel være forståelige nå.

>> plot(x,y,'linewidth',3)

Grafen får tykkelsen 3 ("default"2 er 1).

>> set(gca,'fontsize',20)
```

<sup>2</sup>Med "default" menes her den verdien som blir satt dersom det ikke er spesifisert.

Justerer opp skriftsstørrelsen på aksene.

```
>> xlabel('t [timar]'); ylabel('T [meter]')
```

Setter tekst på aksene (både  $x$ - og  $y$ -aksen).

```
>> hold on
```

Setter at plottene skal bli stående når vi lager flere plott.

```
>> plot([0 24],[-1 -1], 'k-')
```

Plotter linja  $y = -1$  for  $x$  fra og med 0 til og med 24. Linja skal være en svart strek.

```
>> plot([6 18],[3.2 3.2], 'ro', 'linewidth', 2)
>> plot([2.39 9.61 14.39 21.61], -ones(1,4), 'gd')
```

Her plotter vi inn diverse punkt - de første to blir markert med en (tykk) rød (r) sirkel (o), og de fire neste blir markert med grønne (g) “diamanter” (d).

```
>> legend('T(t)', 'y=-1', 'Flo', 'T(t)=-1')
```

Som vi har sett, gjør `legend`-kommandoen oss i stand til å forklare hva de ulike grafene og punktene representerer. “Input”-variablene til `legend`-, `xlabel` og `ylabel`-funksjonene er små tekst-snutter inni apostroffer<sup>3</sup>. Slike små tekst-variabler kalles *strenger*.

```
>> axis([0 27 -4 4])
```

Her justerer vi  $x$ -aksen litt ut slik at grafen ikke kolliderer med teksten i hjørnet øverst til høyre.

b) Om vi tar med kommentarer, kan skriptet se slik ut:

```
% Skript som plotter en modell for tidevann
x=0:.1:24; % Vektor med argument-verdier
y=3.2*sin(pi/6*(x-3)); % Vektor med fusjonsverdier
plot(x,y, 'k', 'linewidth', 3) % Plotter funksjonen
set(gca, 'fontsize', 20) % Fikserer skriftsstørrelsen på aksene
xlabel('t [timar]'); ylabel('T [meter]') % Setter tekst på aksene
hold on
plot([0 24],[-1 -1], 'k-') % Markerer linja y=-1
plot([6 18],[3.2 3.2], 'ro', 'linewidth', 2) % Markerer punktene for flo
plot([2.39 9.61 14.39 21.61], -ones(1,4), 'gd') % Markerer når vannstanden var -1 m
legend('T(t)', 'y=-1', 'Flo', 'T(t)=-1') % Forklarende tekst til grafene
axis([0 27 -4 4]) % Justerer intervallene aksene går over
hold off
```

---

<sup>3</sup>Hermetegn kan også brukes

Det er en god vane å kommentere skript. Her er det kanskje gjort litt vel omstendelig. Dette vil vi komme tilbake til i neste leksjon.