Harmoniske svingninger
og
Sammensat fjederkonstant

Jacob Christensen

Fysiske Undervisnings-Forsøg
Roskilde Universitets Center

21. januar 2008

Harmoniske svingninger og sammensat fjederkonstant

Formål

Formålet med øvelsen er at eftervise formlen for svingningstiden i harmniske svingninger og ved samme lejlighed tjekke additionsreglerne for fjederkonstanter.

Apparatur

- fjedre
- lodder
- vægt
- newtonmeter
- CBL afstandsmåler
- Labpro
- computer med softwaren Loggerpro
- forsøgsstativ

Fremgangsmåde

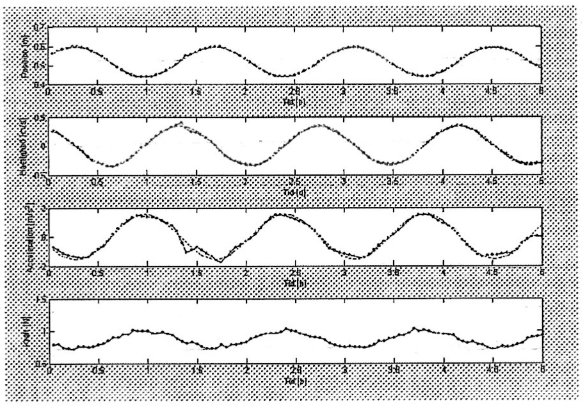
Nogle lodder vejes, så deres masse(incl. ophæng og snavs mm.) er kend med rimelig nøjagtighed. En Labpro tilsluttes newtonmeter og afstandsmåler, så man kan måle kraf og position samtidigt. Newtonmeteret hænges op i et forsøgsstativ og afstandsmåleren lægges nedenunder. Fra newtonmeteret hænges et lod i en eller flere fjedre. Lodet trækkes en smugle ned og slippes så en lodret svingning af loddet igangsættes. En måleserie af samtidig position og kraft startes. Afstandsmåleren leverer ved samme lejlighed en måling af hastighed og acceleration.

Måleresultater

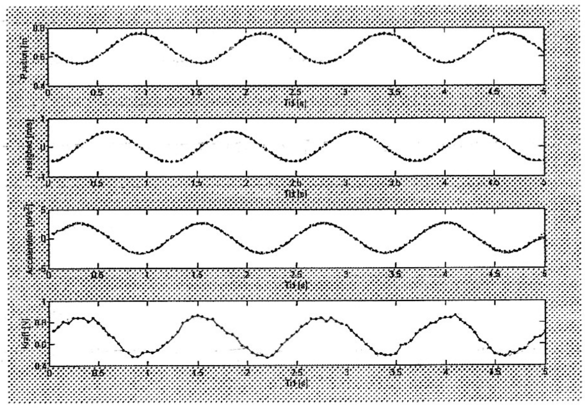
Selve måleresultaterne er for rige på data til at præsenteres i tabel. I stedet præsenteres måleresultaterne i grafer. Tabellen er en guide til at se hvilke lodder, fjedre og opsætning, der er anvendt i de enkelte måleserier.

Serie	Masse	Antal fjedre	Opsætning	Newtonmeter
1	81g	1	-	
2	61g	1	-	
3	61g	2	p	I
4	81g	2	p	
5	31g	2	s	
6	31g	1	-	
7	61g	1	-	
8	81g	1	-	
9	81g	2	p	II
10	61g	2	p	
11	31g	2	p	
12 a+b	31g	2	s	

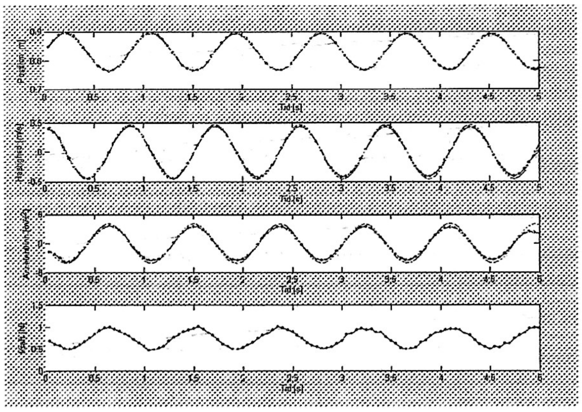
Tabel 1:Oversigt over opsætningen i måleserierne.



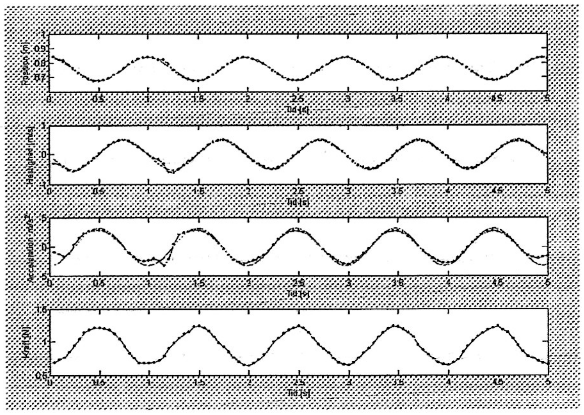
Graf 1: Måleresultater og bedste fit serie 1.



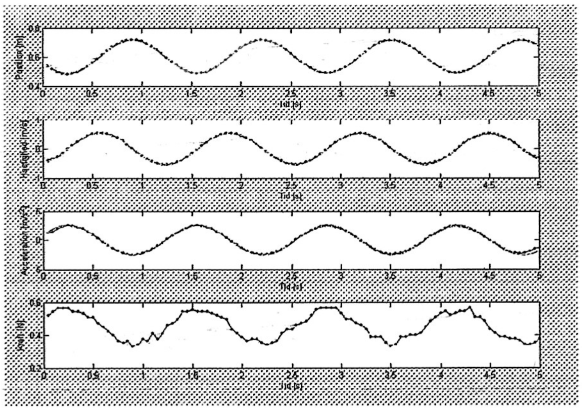
Graf 2: Måleresultater og bedste fit serie 2.



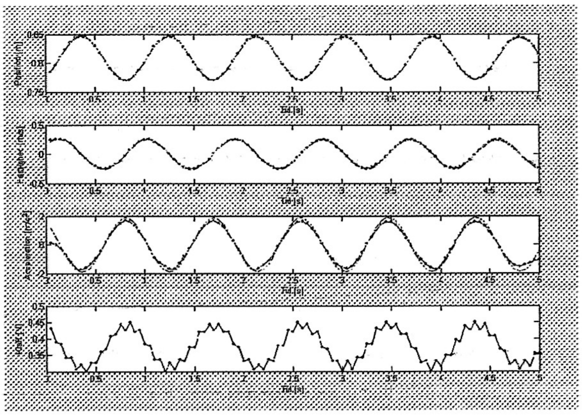
Graf 3: Måleresultater og bedste fit serie 3.



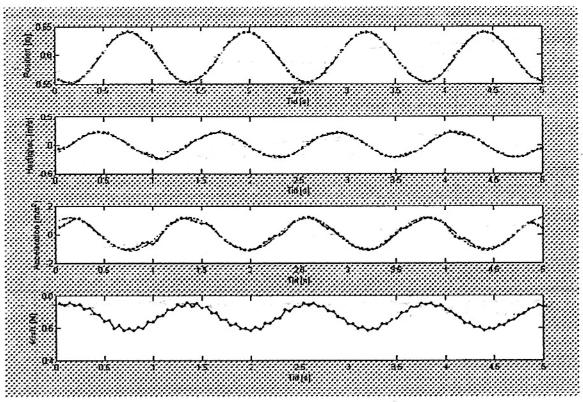
Graf 4: Måleresultater og bedste fit serie 4.



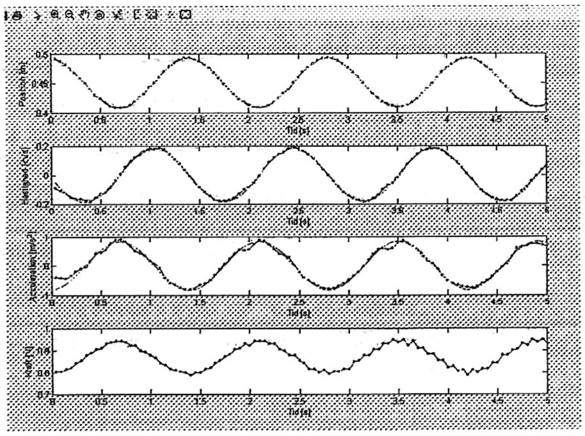
Graf 5: Måleresultater og bedste fit serie 5.



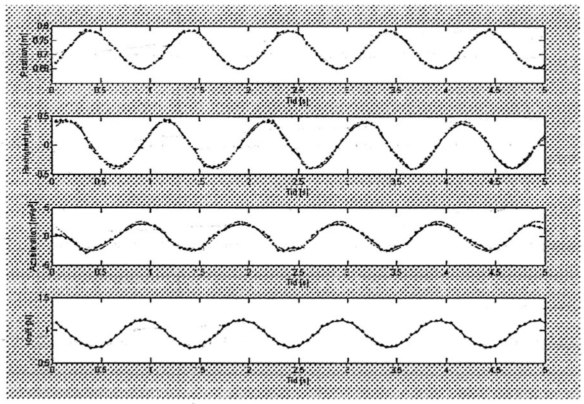
Graf 6: Måleresultater og bedste fit serie 6.



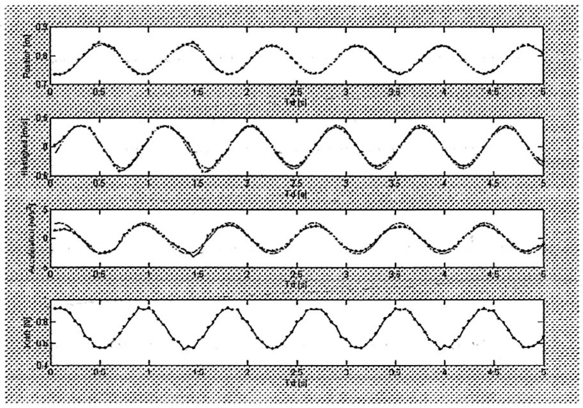
Graf 7: Måleresultater og bedste fit serie 7.



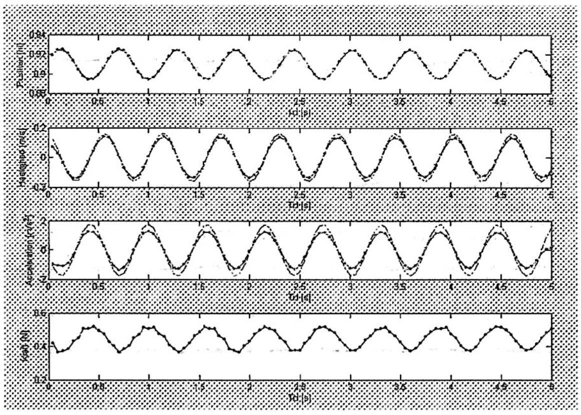
Graf 8: Måleresultater og bedste fit serie 8.



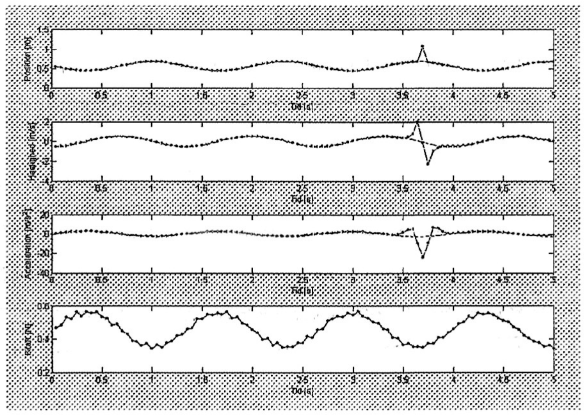
Graf 9: Måleresultater og bedste fit serie 9.



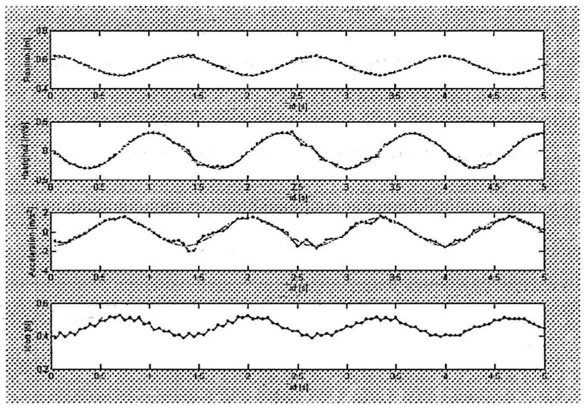
Graf 10: Måleresultater og bedste fit serie 10.



Graf 11: Måleresultater og bedste fit serie 11.



Graf 12: Måleresultater og bedste fit serie 12a.



Graf 13: Måleresultater og bedste fit serie 12b.

Databehandling

En harmonisk svingning er en bevægelse hvor positionen som funktion af tiden kan beskrives ved følgende formel.

$$x(t) = A\ sin(\omega t + \delta) \tag{1}$$

Hvor x er positionen, A er amplituden, ω er vinkelfrekvensen og δ er en fasedrejning. Af formlen (1) kan man finde hastighed og acceleration ved at differentiere en hhv. to gange.

$$x'(t) = \omega A\ cos(\omega t + \delta) \tag{2}$$

$$x''(t) = -\omega^2 A\ sin(\omega t + \delta) \tag{3}$$

Den resulterende kraft F_{res} på et legeme er lig med massen m gange accelerationen.

$$F_{res}(t) = -m\omega^2 A\ sin(\omega t + \delta) \tag{4}$$

Det er sådan indrettet at en fjeder kan levere den kraft der skal til at drive en harmoniske svingning. Kraften fra en udstrakt fjeder er nemlig proportional med udstrækningen og den virker i modsat retning af udstrækningen.

$$F_{fjeder}(t) = -k\ x(t) \tag{5}$$

Når et lod hænger i en fjeder er den resulterende kraft imidlertid summen af fjederkraften og tyngdekraften.

$$F_{res} = F_{fjeder} + F_g$$

Hvor tyndekraften er lig med massen gange tyngdeaccelerationen.

$$F_g = -mg$$

En fjeder kan dog stadig levere den nødvendige kraft til at drive en svingning, hvis man i den harmoniske svingning inkluderer en parallelforskydning A_0.

$$x(t) = A\ sin(\omega t + \delta) + A_0 \tag{6}$$

Paralelforskydningen påvirker ikke hastigheden (2) og accelerationen (3) i bevægelsen.

$$\begin{aligned}
F_{res} &= mx''(x) \\
&= -m\omega^2 A\ sin(\omega t + \delta) \\
&= -m\omega^2 x(t) + m\omega^2 A_0 \\
&= -kx(t) + kA_0 && hvor\ k = m\omega^2 \\
&= -k\ x(t) - mg && hvis\ A_0 = -\frac{mg}{k} \\
&= F_{fjeder} + F_g && hvis\ k\ er\ fjederkonstanten
\end{aligned}$$

Fra ligning(6) kan man finde svingningstiden, da sinus er periodisk med periode 2π. Så for en periode gælder

$$(\omega t_2 + \delta) - (\omega t_1 + \delta) = 2\pi \Rightarrow$$

$$\tau = t_2 - t_1 = \frac{2\pi}{\omega}$$

Man kan isolere ω via. følgende

$$k = m\omega^2 \Rightarrow \omega = \sqrt{\frac{k}{m}}$$

Hvorved man kan formulere svingningstiden τ i termer af massen og fjederkonstanten.

$$\tau = t_2 - t_1 = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{\frac{k}{m}}} = 2\pi\sqrt{\frac{m}{k}} \tag{7}$$

Selve dataopsamlingen foregår i Labpro'en. Der fra kan data overføres til en computer via. softvaren Loggerpro. Man kan lave hele databehandlingen i Loggerpro eller man kan eksportere data til sit favorit matematik-program. Jeg foretrækker matematik-programmet Matlab.
Første opgave i databehandlingen er at fitte modellen(dvs. ligning (6)) til de opsamlede data for hver serie. Når man har et sæt data med begrænset antal datapunkter på en funktion, der er periodisk er det nødvendigt med et manuelt grovfit ellers vil man få problemer med aliasing[1]. Jeg har skrevet en lille stump matlabkode "grovfit"til at vise hvor godt fitteparametrene passer til de observerede data. Når man kan se modellen og data sammen er det forholdsvis let at lave et groft fit. Man skal bare huske på betydningen af de indgående parametre: A styrer udsvingets størrelse, A_0 styrrer den lodrette placering, δ styrrer den vandrette placering og ω styrrer frekvensen. Derefter skal computeren fitte modellen til data. Det gøres ved hjælp af en lille stump kode "errorfun", der giver en talværdi for hvor godt modellen passer til data i form af summen kvadratet på afvigelsere. Denne talværdi minimeres ved hjælp af matlabprogrammet "fminsearch", der kan finde de parametre, der giver den mindste funktionsværdi.

Groft fit:

[1] lige som vognhjul-effekten i gamle film.

Serie	A	ω	δ	A_0
1	0,075	4,5	0,2	0,51
2	0,095	5,1	3,0	0,65
3	0,07	7,3	0	0,84
4	0,077	6,4	1,4	0,75
5	0,11	4,9	3,3	0,61
6	0,037	7,1	-1,	0,81
7	0,042	5,2	3,8	0,595
8	0,042	4,45	1,7	0,45
9	0,066	6,250	-0,92	0,72
10	0,055	7,3	3,9	0,79
11	0,0155	10,9	0	0,91
12a	0,11	4,8	3,0	0,56
12b	0,07	4,8	1,3	0,56

Fint fit:

Serie	A	ω	δ	A_0
1	0,0788	4,4329	0,3629	0,5182
2	0,1034	5,0922	3,1291	0,6562
3	0,0641	7,2795	0,0003	0,8302
4	0,0808	6,3478	1,5466	0,7571
5	0,1147	4,8089	3,4884	0,6040
6	0,0379	7,0721	-1,0004	0,8086
7	0,0437	5,1647	3,8965	0,5957
8	0,0418	4,4672	1,6250	0,4507
9	0,0659	6,2582	-0,9850	0,7150
10	0,0516	7,2793	4,0343	0,7879
11	0,0150	10,8659	0,1541	0,9099
12a	0,1220	4,7574	3,0365	0,5610
12b	0,0645	4,7692	1,3562	0,5556

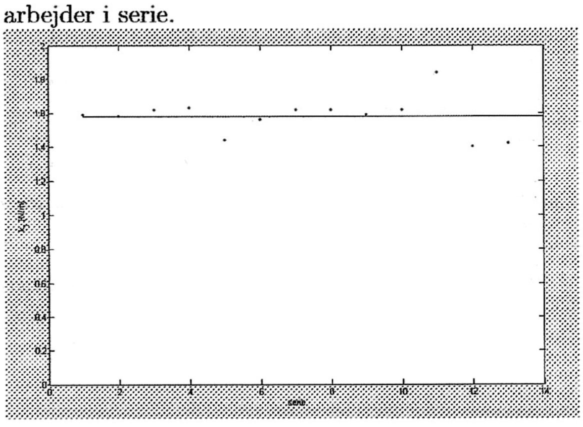
Blandt disse modelparametre er både A og δ helt knyttet til den enkelte måling. Idet A beskriver hvor langt der blev hevet ned i loddet inden det blev sluppet. Og δ beskriver hvornår loddet blev sluppet i forhold til tidstagningens start. Det er altså ikke dér fysikken ligger. A_0 kunne for så vidt bruges til at sige noget om hvor stiv fjederen/fjedrene er. Men hvis man gør det er målingen behæftet med en fejlkilde: Lodder med forskellige masser har også forskellige udformninger, så positionen af bunden af lodderne afhænger ikke kun af massen og fjederkonstanten, men også af loddets form. Til den videre databehandling fokuseres på ω. I det følgende udregnes svingningstiden $\tau = 2\pi/\omega$ og fjederkonstanten for den pågældende opsætning $k = m\omega^2$.

Serie	Masse	Antal fjedre	Opsætning	ω	$\tau = 2\pi/\omega$	$k = m\omega^2$	k_1
#	kg	#		Rad/s	s	N/m	N/m
1	0,081	1	-	4,43	1,41	1,59	1,59
2	0,061	1	-	5,09	1,23	1,58	1,58
3	0,061	2	p	7,28	0,86	3,23	1,62
4	0,081	2	p	6,34	0,99	3,26	1,63
5	0,031	2	s	4,81	1,31	0,72	1,44
6	0,031	1	-	7,07	0,89	1,56	1,56
7	0,061	1	-	5,16	1,22	1,62	1,62
8	0,081	1	-	4,47	1,41	1,62	1,62
9	0,081	2	p	6,26	1,00	3,17	1,59
10	0,061	2	p	7,28	0,86	3,23	1,62
11	0,031	2	p	10,9	0,58	3,68	1,84
12a	0,031	2	s	4,76	1,32	0,70	1,40
12b	0,031	2	s	4,77	1,32	0,71	1,42

Når to eller flere fjedre anvendes samtidig kan deres samlede fjederkonstant k findes ved at lægge de enkelte fjedrekonstanter sammen på følgende vis:

Parallelt:
$k = k_1 + k_2 + \ldots$
Serie:
$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2} + \ldots$

Så hvis to ens fjedre arbejder sammen parallelt bliver den samlede fjederkonstant $k = 2k_1$. Og når de arbejder i serie bliver fjederkonstanten $k = \frac{k_1}{2}$. For at finde fjederkonstanten k_1 for en enkelt fjeder må man altså dividere den samlede fjederkonstant med 2 når fjedrene arbejder parallelt og gange med 2 når de arbejder i serie.



Som det ses af figuren er fjederkonstanten for én fjeder rimeligt konstant. Hvilket er indikator for at svingningstiden er godt beskrevet ved (7).

$$\tau = 2\pi\sqrt{\frac{m}{k}}$$

Fejlkilder

Der er ikke taget højde i hvilken grad de anvendte fjedre er ens. Det er blot antaget at de er helt ens! Der er heller ikke taget højde for fjedrenes egen masse! De to fejlkilder kommer især til udtryk i seriene med de mindste lodder, hvor massen af fjederen bliver sammenlignelig med massen af loddet. Og hvor den lille masse i øvrigt gør målingen følsom over for enhver fejlkilde.

Konklusion

Svingningstiden for en harmonisk ossilator, i form af et lod ophængt i en fjeder, er godt beskrevet ved

$$\tau = 2\pi\sqrt{\frac{m}{k}}$$

Hvor den samlede fjederkonstant k kan findes ved nedenstående, når der ingår to eller flere fjedre.

Parallelt:
$k = k_1 + k_2 + \ldots$
Serie:
$\frac{1}{k} = \frac{1}{k_1} + \frac{1}{k_2} + \ldots$

Appendiks

Matlabfiler

Errorfun

```
function out=errorfun(in,x,y);
k1=in(1);k2=in(2);k3=in(3);k4=in(4);
yy=k1*sin(k2.*x+k3)+k4;
out=sum((y-yy).^2,1);
```

Bearb

```
data
%y(x)=k1*sin(k2.*x+k3)+k4
x=0;
y=0;
grovfit=[
    0.075 4.5 0.2 0.51
    0.095 5.1 3 0.65
    0.07 7.3 0 0.84
    0.077 6.4 1.4 0.75
    0.11 4.9 3.3 0.61
    0.037 7.1 -1 0.81
    0.042 5.2 3.8 0.595
    0.042 4.45 1.7 0.45
    0.066 6.25 -0.92 0.72
    0.055 7.3 3.9 0.79
    0.0155 10.9 0 0.91
    0.11 4.8 3 0.56
    0.07 4.8 1.3 0.56];
for N=1:9;
    s=['x=serie_00' mat2str(N) '(:,1); y=serie_00' mat2str(N) '(:,3);'];
    eval(s);
    pause(0.1);
    coef(N,:)=fminsearch(@(inp) errorfun(inp,x,y),grovfit(N,:));
end
```

```
for N=10:11;
    s=['x=serie_0' mat2str(N) '(:,1); y=serie_0' mat2str(N) '(:,3);'];
    eval(s)
    coef(N,:)=fminsearch(@(inp) errorfun(inp,x,y),grovfit(N,:));
end
x=serie_012a(:,1); y=serie_012a(:,3);
coef(12,:)=fminsearch(@(inp) errorfun(inp,x,y),grovfit(12,:));
x=serie_012b(:,1); y=serie_012b(:,3);
coef(13,:)=fminsearch(@(inp) errorfun(inp,x,y),grovfit(13,:));

winsize=[1 31 1024 666];
for N=1:9;
    s=['x=serie_00' mat2str(N) '(:,1);y=serie_00' mat2str(N) '(:,3);'];
    eval(s);
    s=['v=serie_00' mat2str(N) '(:,4);a=serie_00' mat2str(N) '(:,5);F=serie_00' mat2str(N)
    eval(s);
    figure(N)
    set(figure(N),'position',winsize);

    yy=coef(N,1).*sin(coef(N,2).*x+coef(N,3))+coef(N,4);
    vv=coef(N,2).*coef(N,1).*cos(coef(N,2).*x+coef(N,3));
    aa=-coef(N,2).^2.*coef(N,1).*sin(coef(N,2).*x+coef(N,3));
    s=['Serie ' mat2str(N)];
    subplot(4,1,1)
    title(s)
    plot(x,y,'b.-',x,yy,'r-');
    xlabel('Tid [s]')
    ylabel('Position [m]')
    subplot(4,1,2)
    plot(x,v,'b.-',x,vv,'r-');
    xlabel('Tid [s]')
    ylabel('Hastighed [m/s]')
    subplot(4,1,3)
    plot(x,a,'b.-',x,aa,'r-')
    xlabel('Tid [s]')
    ylabel('Acceleration [m/s^2]')
    subplot(4,1,4)
    plot(x,F,'k.-');
    xlabel('Tid [s]')
    ylabel('Kraft [N]')

end
for N=10:11;
    s=['x=serie_0' mat2str(N) '(:,1);y=serie_0' mat2str(N) '(:,3);'];
    eval(s);
    s=['v=serie_0' mat2str(N) '(:,4);a=serie_0' mat2str(N) '(:,5);F=serie_0' mat2str(N) '(:
```

```
    eval(s);
    figure(N)
    set(figure(N),'position',winsize);

    yy=coef(N,1).*sin(coef(N,2).*x+coef(N,3))+coef(N,4);
    vv=coef(N,2).*coef(N,1).*cos(coef(N,2).*x+coef(N,3));
    aa=-coef(N,2).^2.*coef(N,1).*sin(coef(N,2).*x+coef(N,3));

    s=['Serie ' mat2str(N)];
    subplot(4,1,1)
    title(s)
    plot(x,y,'b.-',x,yy,'r-');
    xlabel('Tid [s]')
    ylabel('Position [m]')
    subplot(4,1,2)
    plot(x,v,'b.-',x,vv,'r-');
    xlabel('Tid [s]')
    ylabel('Hastighed [m/s]')
    subplot(4,1,3)
    plot(x,a,'b.-',x,aa,'r-')
    xlabel('Tid [s]')
    ylabel('Acceleration [m/s^2]')
    subplot(4,1,4)
    plot(x,F,'k.-');
    xlabel('Tid [s]')
    ylabel('Kraft [N]')
end
N=12;
x=serie_012a(:,1);y=serie_012a(:,3);
v=serie_012a(:,4);a=serie_012a(:,5);
F=serie_012a(:,2);
figure(N)
set(figure(N),'position',winsize);

yy=coef(N,1).*sin(coef(N,2).*x+coef(N,3))+coef(N,4);
vv=coef(N,2).*coef(N,1).*cos(coef(N,2).*x+coef(N,3));
aa=-coef(N,2).^2.*coef(N,1).*sin(coef(N,2).*x+coef(N,3));

s=['Serie 12a'];
subplot(4,1,1)
title(s)
plot(x,y,'b.-',x,yy,'r-');
xlabel('Tid [s]')
ylabel('Position [m]')
subplot(4,1,2)
plot(x,v,'b.-',x,vv,'r-');
```

```
xlabel('Tid [s]')
ylabel('Hastighed [m/s]')
subplot(4,1,3)
plot(x,a,'b.-',x,aa,'r-')
xlabel('Tid [s]')
ylabel('Acceleration [m/s^2]')
subplot(4,1,4)
plot(x,F,'k.-');
xlabel('Tid [s]')
ylabel('Kraft [N]')


N=13;
x=serie_012b(:,1);y=serie_012b(:,3);
v=serie_012b(:,4);a=serie_012b(:,5);
F=serie_012b(:,2);
figure(N)
set(figure(N),'position',winsize);

yy=coef(N,1).*sin(coef(N,2).*x+coef(N,3))+coef(N,4);
vv=coef(N,2).*coef(N,1).*cos(coef(N,2).*x+coef(N,3));
aa=-coef(N,2).^2.*coef(N,1).*sin(coef(N,2).*x+coef(N,3));

s=['Serie 12b'];
subplot(4,1,1)
title(s)
plot(x,y,'b.-',x,yy,'r-');
xlabel('Tid [s]')
ylabel('Position [m]')
subplot(4,1,2)
plot(x,v,'b.-',x,vv,'r-');
xlabel('Tid [s]')
ylabel('Hastighed [m/s]')
subplot(4,1,3)
plot(x,a,'b.-',x,aa,'r-')
xlabel('Tid [s]')
ylabel('Acceleration [m/s^2]')
subplot(4,1,4)
plot(x,F,'k.-');
xlabel('Tid [s]')
ylabel('Kraft [N]')
```

Grovfit

```
N=11;
coef=[0.11 4.8 3 0.56];
```

```
%s=['x=serie_0' mat2str(N) '(:,1);y=serie_0' mat2str(N) '(:,3);'];
%eval(s);
%s=['v=serie_0' mat2str(N) '(:,4);a=serie_0' mat2str(N) '(:,5);F=serie_0' mat2str(N) '(:,2).
%eval(s);
x=serie_012a(:,1);y=serie_012a(:,3);
v=serie_012a(:,4);a=serie_012a(:,5);
F=serie_012a(:,2);
figure(1)
set(figure(1),'position',winsize);

yy=coef(1).*sin(coef(2).*x+coef(3))+coef(4);
vv=coef(2).*coef(1).*cos(coef(2).*x+coef(3));
aa=-coef(2).^2.*coef(1).*sin(coef(2).*x+coef(3));

subplot(4,1,1)
plot(x,y,'b.-',x,yy,'r-');
subplot(4,1,2)
plot(x,v,'b.-',x,vv,'r-');
subplot(4,1,3)
plot(x,a,'b.-',x,aa,'r-')
subplot(4,1,4)
plot(x,F,'k-');
```