
Matematikk 1000

Øvingsoppgaver i numerikk – leksjon 6

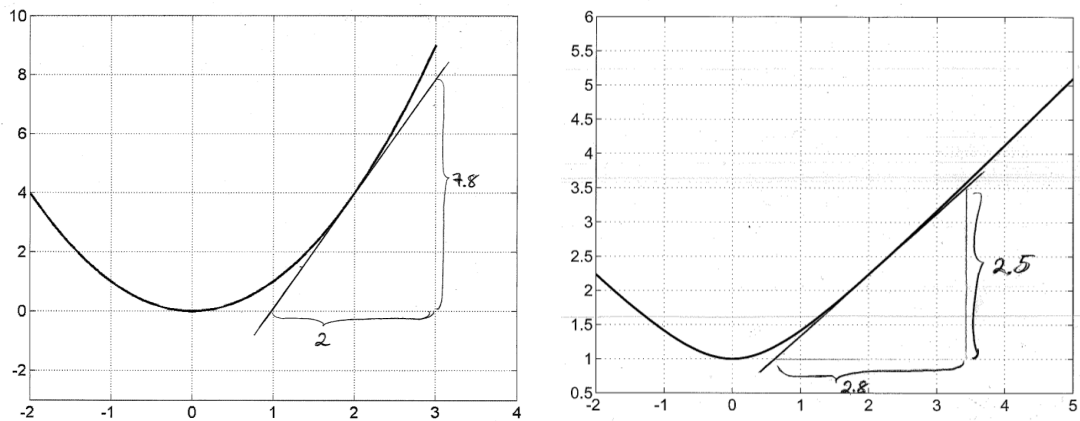
Løsningsforslag

Oppgave 1 – Funksjoner og tangenter

a) Vi plotter grafen med et rutenett:

```
>> x=-3:.1:3;  
>> y=x.^2;  
>> plot(x,y)  
>> grid on  
>> axis([-2 3 -3 10])
```

Vi prøver å tegne inn en tangent for $x = 2$. Dette har vi illustrert i figur 1.



Figur 1: Her har vi plotta grafen til funksjonene i oppgave 1 og forsøkt å tegne inn tangentene i punktet $(2, f(2))$.

Ut fra figuren, kan det set ut til at stigningstallet til tangenten er

$$\frac{7.8 - 0}{3 - 1} = \underline{3.9} \quad .$$

Dette stemte jo ganske bra med det eksakte svaret 4:

$$\begin{aligned}f(x) &= x^2 \\f(2) &= 2^2 = 4 \\f'(x) &= 2x \\f'(2) &= 2 \cdot 2 = 4\end{aligned}$$

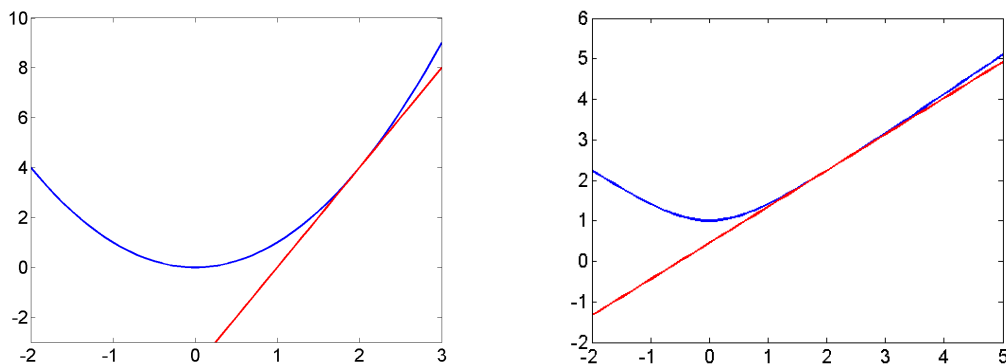
Likninga for tangenten kan vi finne ved formelen

$$y - y_0 = a(x - x_0) \quad ,$$

der a er stigningsallet til linja og (x_0, y_0) er et punkt på linja. Vi får:
 $y - 4 = 4(x - 2) \Leftrightarrow y = 4x - 4$. Vi plottes denne sammen med grafen vi hadde fra før:

```
>> tang=4*x-4;
>> hold on
>> plot(x,tang,'r')
>> hold off
>> axis([-2 3 -3 10])
```

Grafen til f og tangenten i $(2, 4)$ er vist til venstre i figur 2.



Figur 2: Plott for oppgave 1. Den blå kurva viser $f(x)$ og den røde linja er tangenten. Figure til venstre er for deloppgave a), og den til høyre er for deloppgave b). Skriftstørrelsen og tykkelsen på grafene er noe justert i forhold til de kommandoene som er lista opp i teksten.

b)

$$\begin{aligned}f(x) &= \sqrt{1+x^2} \\f(2) &= \sqrt{1+2^2} = \sqrt{5} \\f'(x) &= \frac{1}{2\sqrt{1+x^2}} \cdot (1+x^2)' = \frac{x}{\sqrt{1+x^2}} \\f'(2) &= \frac{2}{\sqrt{1+2^2}} = \frac{2}{\sqrt{5}}\end{aligned}$$

Likning for tangent:

$$y - \sqrt{5} = \frac{2}{\sqrt{5}}(x - 2) \Leftrightarrow y = \frac{2}{\sqrt{5}}x + \frac{1}{\sqrt{5}} \quad .$$

I MATLAB har jeg valgt å plotte grafene for $x \in [-2, 3]$:

```
>> x=-2: .1:5;
>> y=sqrt(1+x.^2);
>> tang=2/sqrt(5)*x+1/sqrt(5);
>> plot(x,y)
>> hold on
>> plot(x,tang,'r')
>> hold off
>> axis([-2 5 -2 6])
```

Grafen til f og tangenten er vist til høyre i figur 2.

Også her har jeg forsøkt å estimere den deriverte ved å finne stigningstallet til en tangent som jeg har tegna inn “for hånd”, se plottet til høyre i figur 1. Her får vi

$$f'(2) \approx \frac{2.5}{2.8} = 0.893 \quad ,$$

noe som stemmer ganske bra med den eksakte verdien $2/\sqrt{5} \approx 0.8944$.

Oppgave 2 – Numerisk derivasjon

a) Jeg har valgt med denne funksjonen:

$$f(x) = \sin x^2 \quad .$$

Vi bruker kjerneregelen når vi deriverer:

$$f'(x) = \cos x^2 \cdot (x^2)' = 2x \cos x^2 \quad .$$

Jeg velger $a = 1$ og får at $f'(a) = f'(1) = 2 \cos 1$.

b) Det vil være nyttig å lage ei funksjonsfil for funksjonen vår:

```
function F=FunkTilOppg2(x)
```

```
% Funksjonen f(x) = sin x^2. Funksjonen tar vektor-argument.
```

```
F=sin(x.^2);
```

Jeg har valgt å kalle denne FunkTilOppg2.m. Vi regner ut gjennomsnittlig vekstrate for noen verdier av h :

```
>> h=1;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = -1.5983
>> h=.5;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = -0.12680
>> h=.1;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 0.94145
>> h=.05;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0174
>> h=1e-2;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0689
>> h=1e-3;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0795
>> h=1e-4;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0805
```

Her kan det være verd å minne om at piltastene er svært nyttige når man skal skrive den samme (lange) kommandoen flere ganger – slik som her. I følge MATLAB er $f'(1) = 2 \cdot \cos 1 \approx 1.0806$, så vi begynner å nærme oss.

c) Vi gjentar det vi nettop gjore med “bakover-formelen”:

```
>> h=1;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 0.84147
>> h=.5;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.1881
>> h=.1;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.1718
>> h=.05;
```

```

>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.1318
>> h=1e-2;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.0918
>> h=1e-3;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.0817
>> h=1e-4;
>> Rate=(FunkTilOppg2(1)-FunkTilOppg2(1-h))/h
Rate = 1.0807

```

Nok en gang ser vi at `Rate` nærmer seg $f'(1)$ når h blir liten.

d) Gjennomsnittet blir

$$\frac{1}{2} \left(\frac{f(a+h) - f(a)}{h} + \frac{f(a) - f(a-h)}{h} \right) = \frac{f(a+h) - f(a) + f(a) - f(a-h)}{2h} = \frac{f(a+h) - f(a-h)}{2h}$$

Altså:

$$f'(a) \approx \frac{f(a+h) - f(a-h)}{2h}$$

når h er passe liten¹. Legg merke til at dette $f'(a)$ -estimatet er helt uavhengig av $f(a)$.

e) Vi gjentar prosedyren fra c) og d) med denne nye formelen:

```

>> h=1;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = -0.37840
>> h=.5;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = f 0.53067
>> h=.1;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = 1.0566
>> h=.05;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = 1.0746
>> h=1e-2;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = 1.0804
>> h=1e-3;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1-h))/2/h
Rate = 1.0806

```

¹“Passe liten” høres unektelig noe upresist ut. Det er mulig å sette opp rimelig presise kriterier for hvor liten h må være for at estimatet vårt skal ha for å ha en bestemt nøyaktighet. Dette skal vi komme tilbake til når vi har lært om Taylor-polynomer.

Det ser ut til at vi nærmer oss riktig verdi for $f'(a)$ raskere på denne måten; for $h = 10^{-3}$ ser vi faktisk ikke forskjell med det antallet desimaler som blir skrevet til skjerm².

f) Maskinpresisjonen finner vi slik:

```
>> eps
ans = 2.2204e-016
```

–altså $2.22 \cdot 10^{-16}$. Dette er jo et veldig lite tall, men det er ikke null. Vi velger noen veldig små verdier for h og beregner den deriverte med “framover-formelen”:

```
>> DerivFasit=2*cos(1)
DerivFasit = 1.0806
>> h=1e-13;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0802
>> h=1e-14;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.0769
>> h=1e-15;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 1.2212
>> h=5e-16;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 0.88818
>> h=1e-16
h = 1.0000e-016
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 0
>> h=5e-17;
>> Rate=(FunkTilOppg2(1+h)-FunkTilOppg2(1))/h
Rate = 0
```

Nå ser vi faktisk at jo mindre h blir, jo *dårligere* estimat får vi for $f'(a)$. Her ser vi en vesens-forskjell mellom numeriske metoder og analytisk matematikk; matematisk skal $(f(x+h) - f(x))/h - f'(a)$ vilkårløst nærme seg 0 når h nærmer seg 0, mens *numerisk* sett blir den deriverte best estimert med en liten – men *endelig* – h . Dette skyldes at datamaskiner har en nedre grense for hvor små tall den klarer å håndtere.

g) Vi kan lage en h -vektor som blir mindre og mindre ved å starte på 1 og så dele på to – igjen og igjen. Dette kan vi gjøre slik i MATLAB:

²MATLAB opererer som tidligere nevnt med langt flere desimaler egentlig; om du vil ha flere desimaler skrevet til skjerm, kan du skrive ‘format long’

```
>> hVekt=[1 .5 .5^2 .5^3 .5^4 .5^5]
hVekt =
    1.000000    0.500000    0.250000    0.125000    0.062500    0.031250
```

Dette kunne også blitt gjort litt mer effektivt:

```
>> n=0:5;
>> hVekt=.5.^n;
```

Vi lager vektorer som beregner $f'(a)$ ut fra de tre metodene over:

```
>> DerivFram=(FunkTilOppg2(1+hVekt)-FunkTilOppg2(1))./hVekt;
>> DerivBak=(FunkTilOppg2(1)-FunkTilOppg2(1-hVekt))./hVekt;
>> DerivMidt=(FunkTilOppg2(1+hVekt)-FunkTilOppg2(1-hVekt))./(2*hVekt);
```

Vi plotter feilen i estimatet ved hjelp av `semilogx`-kommandoen:

```
>> DerivFasit=2*cos(1)
DerivFasit = 1.0806
>> semilogx(hVekt,DerivFram-DerivFasit,'b','linewidth',2)
>> hold on
>> semilogx(hVekt,DerivBak-DerivFasit,'r','linewidth',2)
>> semilogx(hVekt,DerivMidt-DerivFasit,'m','linewidth',2)
>> hold off
>> legend('Framover-formel','Bakover-formel','Midtpunktsformel')
>> axis([.02 1.5 -3 .5])
```

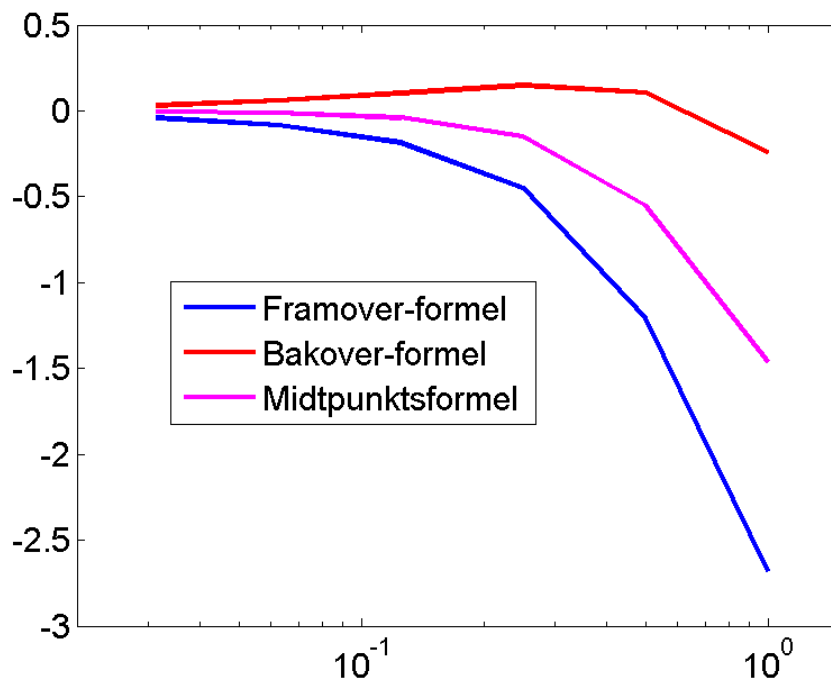
Langs y -aksen har vi her differansen mellom derivasjons-estimatet og den riktige verdien for den deriverte. Vi ser i figur 3 at alle tre metodene gir et estimat med en feil som ser ut til å gå mot 0 når h blir mindre og mindre. Men plottet for midtpunktsformelen går klart mot null raskest. (Her er det nok enklest å lese plottet fra høyre mot venstre.)

I figur 4 har vi plotta absoluttverdien av feilen i de tre ulike måtene å estimere $f'(1)$ på for et veldig stort område av h -verdier. Her er både x - og y -aksen logaritmiske. Vi ser feilen først avtar når h avtar – og at den begynner å vokse igjen når h blir ekstremt liten. Vi ser også at feilen avtar raskere for midtpunktsmetoden enn for framover- og bakovermetoden – og at feilen når en lavere verdi med midtpunktsmetoden enn med de andre formlene. Legg også merke til at grafen begynner å bli “hakkete” når den numeriske feilen til datamaskina begynner å spille en rolle.

- h) Vi kan for eksempel velge $g(x) = 2x^2 + x - 1$. Vi lager ei funksjonfil for denne:

```
function G=AndreGradPol(x)

% AndreGradPol(x) gir funksjonen
```



Figur 3: Plott som viser feilen i ulike estimat for $f'(a)$ som funksjon av h . Plottet er *semi-logaritmisk*. Den blå kurva tilsvarer “framover-formelen”, den røde tilsvarer “bakover-formelen” og den lilla er “midtpunktsformelen”. Se tekst for detaljer.

```
% g(x) = 2x^2+x-1.
% Funksjonen tar både skalar- og
% vektorargumenter.
```

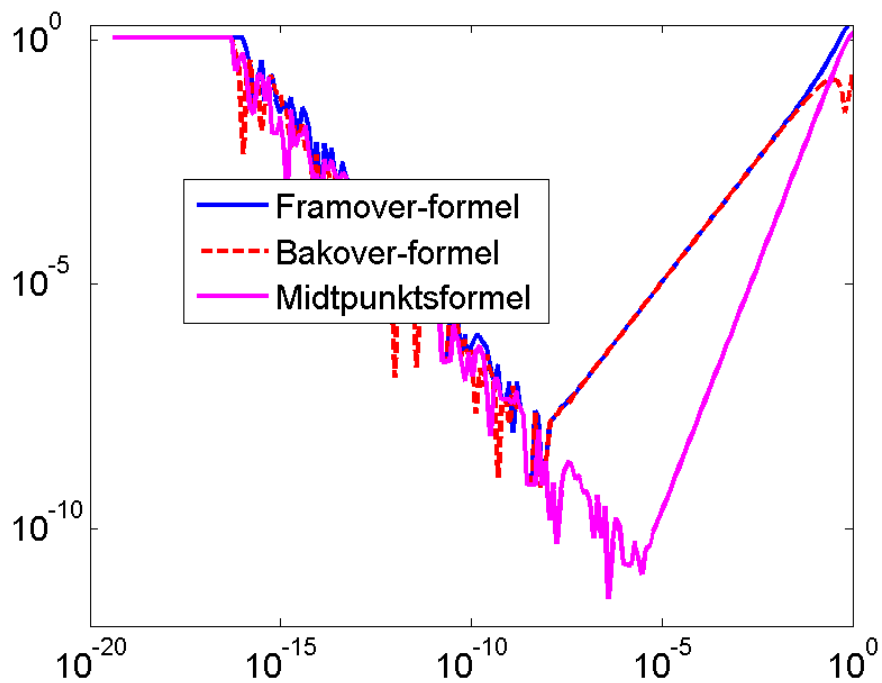
```
G=2*x.^2+x-1;
```

Vi regner noen deriverte:

$$\begin{aligned}
 g'(x) &= 4x + 1 \\
 g'(-1) &= -4 + 1 = -3 \\
 g'(0) &= 1 \\
 g'(2) &= 4 \cdot 2 + 1 = 9 \quad .
 \end{aligned}$$

Vi estimerer disse ulike deriverte i MATLAB:

```
>> a=-1;
>> h=.5;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = -3
```



Figur 4: Feilen i de ulike estimatene for $f'(a)$ for h -verdier fra 1 ned mot 10^{-20} .

```
>> h=.1;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = -3.0000
>> a=0;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = 1.0000
>> h=2;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = 1
>> a=2;
>> h=.5;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = 9
>> h=500;
>> gDeriv=(AndreGradPol(a+h)-AndreGradPol(a-h))/(2*h)
gDeriv = 9
```

Vi ser ut til å få helt nøyaktige svar hver gang – uavhengig av hvilken verdi vi velger for h . Om vi bruker midtpunktsformelen for å finne den deriverte

for en vilkårlig x , ser vi at vi faktisk får $g'(x)$ eksakt:

$$\begin{aligned} \frac{g(x+h) - g(x-h)}{2h} &= \\ \frac{2(x+h)^2 + (x+h) - 1 - (2(x-h)^2 + (x-h) - 1)}{2h} &= \\ \frac{2(x^2 + 2hx + h^2) + x + h - 1 - 2(x^2 - 2hx + h^2 - x + h - 1)}{2h} &= \\ \frac{2x^2 - 2x^2 + 4hx + 4hx + x - x + h + h - 1 + 1}{2h} &= \frac{8hx + 2h}{2h} = \\ 4x + 1 = g'(x) \quad . \end{aligned}$$

Det er ikke så vanskelig å vise at det samme gjelder for et generelt andregradspolynom. Altså: Midtpunktsformelen er faktisk *eksakt* for andregradspolynom.

For hvilken type funksjoner er framover- eller bakoverformelen eksakt?

Oppgave 3 – Den deriverte fra en tabell

a) Plottet kan lages slik:

```
>> Aar=1900:10:2000;
>> Bef=[1650 1750 1860 2070 2300 2520 3020 3700 4450 5300 6123];
>> plot(Aar,Bef,'k','linewidth',2)
>> set(gca,'fontsize',15)
>> xlabel('År')
>> ylabel('Befolkning [i antall millioner]')
```

og resultatet kan sees i figur 5.

b) Vi lar $B(t)$ være antall millioner mennesker i verden i år t . Om vi bruker formelen fra 2 c) kan vi for eksempel estimere $B(1910)$ slik

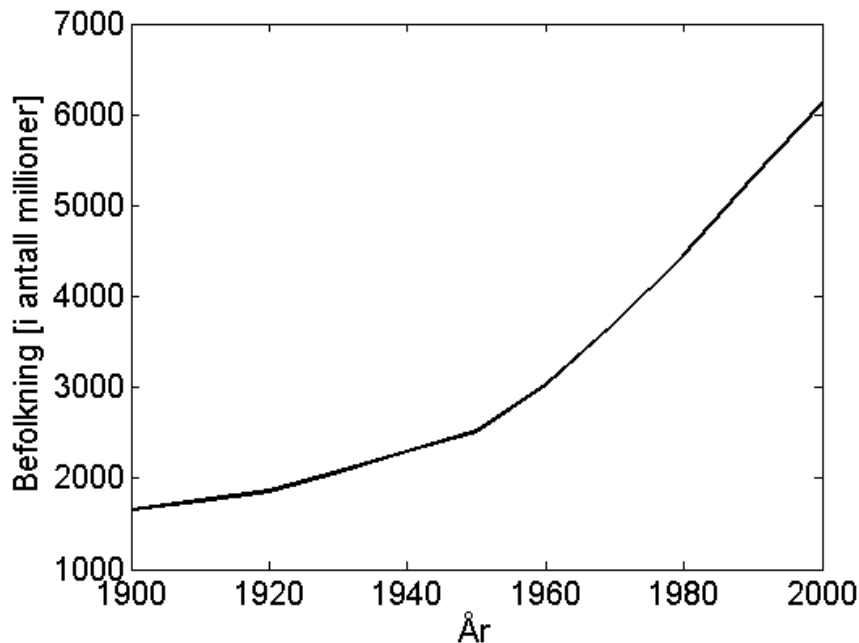
$$B'(1910) \approx \frac{B(1920) - B(1900)}{1920 - 1900} = \frac{B(1910 + 10) - B(1910 - 10)}{2 \cdot 10} \quad .$$

Vi buker MATLAB og får:

```
>> (Bef(3)-Bef(1))/(Aar(3)-Aar(1))
ans = 10.500
```

eller, for å gjøre sammenhengen med midtpunktsformelen litt tydeligere:

```
>> h=Aar(2)-Aar(1);
>> (Bef(3)-Bef(1))/(2*h)
ans = 10.500
```



Figur 5: Verdens befolkning i antall millioner som funksjon av årstall.

Altså, i år 1910 vokste jordas befolkning med ca 10.5 millioner mennesker per år. For de andre årene får vi $B'(1920) \approx 16$, $B'(1930) \approx 22$, $B'(1940) \approx 22.5$, $B'(1950) \approx 36$, $B'(1960) \approx 59$, $B'(1970) \approx 71.5$, $B'(1980) \approx 80$ og $B'(1990) \approx 83.65$.

- c) Siden vi ikke har $B(1890)$, kan vi ikke bruke midpunktsformelen for å estimere $B'(1900)$. På samme måte er vi ute at stand til bruke midpunktsformelen for å finne $B'(2000)$ siden vi ikke har fått oppgitt $B(2010)$. Derimot kan vi bruke “framover”- og “bakover”-formlene selv om de ikke er like nøyaktige:

$$B'(1900) \approx \frac{B(1910) - B(1900)}{1910 - 1900} = 10$$

$$B'(2000) \approx \frac{B(2000) - B(1990)}{2000 - 1990} = 82.3$$

- d) Alle de estimatene vi har regna ut her, kan gjøres i “ett jafs” med dette skriptet:

```

1 % Vektor med årstall:
2 Aar=1900:10:2000;
3 % Vektor med befolkning:
4 Bef=[1650 1750 1860 2070 2300 2520 3020 3700 4450 5300 6123];
5 % Bestemmer lengda av Aar og tilordner dette tallet til N:

```

```

6  N=length(Aar);
7
8  for n=1:N
9      if n==1                                % For det første punktet
10         Bderiv(n)=(Bef(n+1)-Bef(n))/(Aar(n+1)-Aar(n)); % Framover-formel
11     elseif n==N                            % For det siste punktet
12         Bderiv(n)=(Bef(n)-Bef(n-1))/(Aar(n)-Aar(n-1)); % Bakover-formel
13     else                                    % For alle andre punkt
14         Bderiv(n)=(Bef(n+1)-Bef(n-1))/(Aar(n+1)-Aar(n-1)); % Midtpunktsformel
15     end
16 end
17 % Skriver estimatene for de deriverte til skjerm:
18 Bderiv
19
20 % Plotter vekstfarten
21 figure(1)
22 plot(Aar,Bderiv,'k-','linewidth',2)
23 set(gca,'fontsize',15)
24 xlabel('År')
25 ylabel('Vekstfart [millioner per år]')
26
27 % Plotter den relative vekstfarten
28 figure(2)
29 plot(Aar,Bderiv./Bef*100,'k-','linewidth',2)
30 set(gca,'fontsize',15)
31 xlabel('År')
32 ylabel('Relativ evkstfart [prosent]')

```

Vi har kalt skriptet BefDeriv.m. Vi kjører det i MATLAB:

```

>> BefDeriv
Bderiv =

```

Columns 1 through 8:

```

    10.000    10.500    16.000    22.000    22.500    36.000    59.000    71.500

```

Columns 9 through 11:

```

    80.000    83.650    82.300

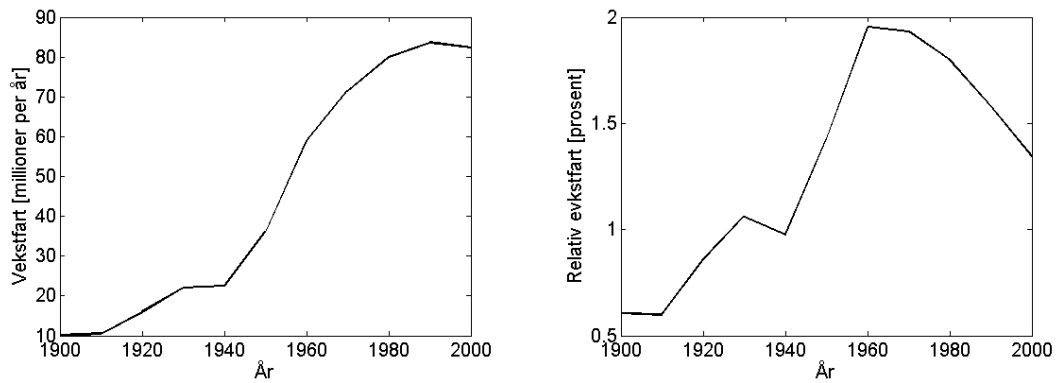
```

Vi får også opp plottene vist i figur 6. Som sagt, kunne dette også vært gjort uten å bruke noen if-sats. Linjene fra og med nummer 8 til og med 16 kan erstattes med følgende:

```

% For-løkke for alle indre punkt
for n=2:(N-1)

```



Figur 6: Tilnærma årlig tilvekst i verdens befolkning som funksjon av årstall.
 Figure til venstre gir den relative tilveksten, altså tilvekst i forhold til befolkning,
 i prosent.

```
Bderiv(n)=(Bef(n+1)-Bef(n-1))/(Aar(n+1)-Aar(n-1)); % Midtpunktsformel
end
% Endepunktene:
Bderiv(1)=(Bef(2)-Bef(1))/(Aar(2)-Aar(1));
Bderiv(N)=(Bef(N)-Bef(N-1))/(Aar(N)-Aar(N-1));
```

Denne varianten er noe ryddigere siden vi ikke trenger bruke noen `if`-
 satser.