

Numerisk derivasjon

Anne Kværnø

Problemstilling

Gitt en tilstrekkelig glatt funksjon f . Finn en tilnærmelse til $f'(x)$ i et gitt punkt x .

Den deriverte av f (https://wiki.math.ntnu.no/tma4100/tema/differentiation?&#definisjonen_av_den_deriverte_gitt_som_en_grenseverdi), er definert ved:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}.$$

Dette kan selvfølgelig brukes direkte som en numerisk tilnærmelse til den deriverte i et gitt punkt x .

Under er et lite knippe vanlige tilnærmelser til den deriverte. For $h > 0$ tilstrekkelig liten:

$$f'(x) \approx \begin{cases} \frac{f(x+h)-f(x)}{h} & \text{Foroverdifferanse} \\ \frac{f(x)-f(x-h)}{h} & \text{Bakoverdifferanse} \\ \frac{f(x+h)-f(x-h)}{2h} & \text{Sentraldifferanse} \end{cases}$$

Den første av disse er hentet direkte fra definisjonen, det er for såvidt også den andre, mens den tredje rett og slett er snittet av disse.

Hvor nøyaktige er disse tilnærmelsene? Hvordan finner vi slike differanseformler? Hvordan velge h ? Dette er ting vi skal se på i dette notatet. Senere skal vi også se hvordan disse differanseformlene blir brukt for å løse andre ordens differensialligningen der løsningen er kjent i de to randpunktene, og til å løse partielle differensialligninger numerisk.

La oss starte med å implementere og teste de tre differanseformlene gitt over.

In [1]:

```
# Importer nødvendige moduler, og sett parametre for plotting.
# Dette må alltid kjøres først!
%matplotlib inline
from numpy import *           # Matematiske funksjoner og lin.alg.
from numpy.linalg import norm, solve
from matplotlib.pyplot import * # Grafikk
newparams = {'figure.figsize': (8.0, 4.0), 'axes.grid': True,
             'lines.markersize': 8, 'lines.linewidth': 2,
             'font.size': 14}
rcParams.update(newparams)
```

In [2]:

```
# Numerisk derivasjon vha. foroverdifferanse
def diff_forover(f, x, h=0.1):
    return (f(x+h)-f(x))/h

# Numerisk derivasjon vha. foroverdifferanse
def diff_bakover(f, x, h=0.1):
    return (f(x)-f(x-h))/h

# Numerisk derivasjon vha. sentraldifferanse
def diff_sentral(f, x, h=0.1):
    return (f(x+h)-f(x-h))/(2*h)
```

Eksempel

Test de tre metodene på $f(x) = \sin(x)$ i $x = \pi/4$. Sammenlign med eksakt løsning, $\cos(\pi/4) = \sqrt{2}/2$. Prøv med forskjellige valg av h , f.eks. $h = 0.1, h = 0.01, h = 0.001$. Legg merke til hvordan feilen $e(x; h)$ endrer seg med h .

In [3]:

```
x = pi/4;
df_eksakt = cos(x)
h = 0.001
f = sin
df = diff_forover(f, x, h)
print('Foroverdifferanse: df = {:.12.8f},    Feil = {:.10.3e} '.format(df, df_eksakt-df))
df = diff_bakover(f, x, h)
print('Bakoverdifferanse: df = {:.12.8f},    Feil = {:.10.3e} '.format(df, df_eksakt-df))
df = diff_sentral(f, x, h)
print('Sentraldifferanse: df = {:.12.8f},    Feil = {:.10.3e} '.format(df, df_eksakt-df))
```

```
Foroverdifferanse: df =    0.70675311,    Feil =    3.537e-04
Bakoverdifferanse: df =    0.70746022,    Feil =   -3.534e-04
Sentraldifferanse: df =    0.70710666,    Feil =    1.179e-07
```

Feilanalyse

I dette tilfellet er feilanalysen ganske enkelt: Gjør en Taylorutvikling (<https://wiki.math.ntnu.no/tma4100/tema/taylorpolynomials>) av feilen rundt x og se hvor mange ledd som forsvinner. Vi må forutsette at f er tilstrekkelig kontinuerlig deriverbar.

Forover- og bakoverdifferanser:

$$e(x; h) = f'(x) - \frac{f(x+h) - f(x)}{h} = f'(x) - \frac{f(x) + f'(x)h + \frac{1}{2}f''(\xi)h^2 - f(x)}{h} = -\frac{1}{2}f''(\xi)h$$

der ξ ligger mellom x og $x+h$.

Tilsvarende vil feilen for bakoverdifferansen være:

$$e(x; h) = \frac{1}{2}f''(\xi)h$$

Sentraldifferansen:

$$\begin{aligned} e(x; h) &= f'(x) - \frac{f(x+h) - f(x-h)}{2h} \\ &= f'(x) - \frac{\left(f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(\xi_1)h^3\right) - \left(f(x) - f'(x)h + \frac{1}{2}f''(x)h^2 - \frac{1}{6}f'''(\xi_2)h^3\right)}{2h} \\ &= -\frac{1}{12}(f'''(\xi_1) + f'''(\xi_2))h^2 \\ &= -\frac{1}{6}f'''(\eta)h^2, \quad \eta \in (x-h, x+h). \end{aligned}$$

Det er to Taylorutviklinger med restledd, slik at $\xi_1 \in (x, x+h)$ og $\xi_2 \in (x-h, x)$. De to restleddene kan slås sammen ved bruk av den generaliserte middelverdisetningen.

Hvis Taylor-utviklingen av feilen $e(x, h) = C_p h^p + C_{p+1} h^{p+1} + \dots$ der $C_p \neq 0$ sier vi at differanseformelen er av *orden* p . Forover- og bakoverdifferansen er altså av orden 1, sentraldifferansen av orden 2.

Bruk av differanseformler.

Grovt sett kan vi tenke oss to situasjoner:

1. Funksjonen $f(x)$ kan beregnes i alle punkter, så h kan velges fritt. Det vil f.eks. være tilfelle hvis vi ønsker å bruke disse tilnærmelsene for å finne en numerisk tilnærmelse til Jakobi-matrisen i Newtons metode for å løse ikke-lineære ligninger.
2. Funksjonen $f(x)$ (eller tilnærmelser til denne) er bare kjent for enkelte verdier av x . Eksempel på dette vil være når disse brukes for å løse partielle differensialligninger.

Hvordan lage differanseformler:

Sett av vi ønsker å finne en tilnærmelse til $f'(x)$ i et gitt punkt x basert på verdien av f beregnet i 3 distinkte punkter: x , $x + h_1$ og $x + h_2$. For enkelhets skyld, sett $h_1 = \theta_1 h$ og $h_2 = \theta_2 h$, hvor $\theta_1 = \pm 1$.

Differanseformelen vil bli på formen

$$Df = af(x) + bf(x + \theta_1 h) + cf(x + \theta_2 h).$$

Oppgaven blir da å finne konstantene a , b og c slik at flest mulig ledd i Taylorutviklingen av feilen blir 0:

$$\begin{aligned} e(x; h) &= f'(x) - af(x) - bf(x + \theta_1 h) - cf(x + \theta_2 h) \\ &= f'(x) - af(x) \\ &\quad - b \left(f(x) + \theta_1 h f'(x) + \frac{1}{2} \theta_1^2 h^2 f''(x) + \frac{1}{6} \theta_1^3 h^3 f'''(x) + \dots \right) \\ &\quad - c \left(f(x) + \theta_2 h f'(x) + \frac{1}{2} \theta_2^2 h^2 f''(x) + \frac{1}{6} \theta_2^3 h^3 f'''(x) + \dots \right) \end{aligned}$$

Sammenlign like ledd ($f(x)$, $f'(x)$, $f''(x)$, osv). Vi ønsker flest mulig av disse leddene lik 0:

$$\begin{aligned} f(x) : & \quad -a - b - c = 0 \\ f'(x) : & \quad 1 - \theta_1 h b - \theta_2 h c = 0 \\ f''(x) : & \quad -\frac{1}{2} \theta_1^2 h^2 b - \frac{1}{2} \theta_2^2 h^2 c = 0 \end{aligned}$$

Dette gir oss tre ligninger og tre ukjente, med løsning:

$$a = -\frac{1}{h} \frac{\theta_1 + \theta_2}{\theta_1 \theta_2}, \quad b = \frac{1}{h} \frac{\theta_2}{\theta_1(\theta_2 - \theta_1)}, \quad c = \frac{1}{h} \frac{\theta_1}{\theta_2(\theta_1 - \theta_2)}.$$

Ved å bruke disse verdiene for a , b og c får vi en differanseformel av orden 2, og med feil

$$e(x; h) = -\frac{h^3}{6} (b\theta_1^3 + c\theta_2^3) f'''(x) + \dots = \frac{1}{6} \theta_1 \theta_2 h^2 f'''(x) + \dots$$

Fra dette får vi noen mye brukte differanseformler:

$$\begin{aligned} \theta_1 = 1, \theta_2 = -1 : \quad f'(x) &= \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(x) + \dots \\ \theta_1 = 1, \theta_2 = 2 : \quad f'(x) &= \frac{-f(x+2h) + 4f(x+h) - 3f(x)}{2h} + \frac{h^2}{3} f'''(x) + \dots \\ \theta_1 = -1, \theta_2 = -2 : \quad f'(x) &= \frac{3f(x) - 4f(x-h) + f(x-2h)}{2h} + \frac{h^2}{3} f'''(x) + \dots \end{aligned}$$

Det er mulig å vise at feilleddene faktisk er $\frac{h^2}{6} f'''(\xi)$ for sentraldifferansen og $\frac{h^2}{3} f'''(\xi)$ for de to andre.

Høyere ordens differanseformler

Differanseformler for høyere ordens deriverte kan finnes på samme måte. Den eneste vi kommer til å bruke i dette kurset er sentraldifferansen for $f''(x)$, gitt ved.

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi), \quad \xi \in (x-h, x+h).$$

Avrundingsfeil i differanseformler

Forutsatt at h kan velges fritt er det selvsagt fristende å velge denne svært liten for å få en nesten eksakt tilnærming. Det kan gå galt, som følgende eksperiment viser.

Eksperiment

La $f(x) = \sin(x)$, og finn en tilnærming til $f'(x)$ ved bruk av en av differanseformlene over. Gjør dette for stadig mindre h (i praksis vil vi halvere h). Plott absoluttverdien av feilen som funksjon av h .

Hva forventer vi å observere, og hvordan kan det best illustreres i et plott?

For en metode av orden p vil

$$|e(x; h)| \approx Ch^p.$$

Ta logaritmen på begge sider:

$$\log|e(h)| \approx p \log(h) + \log(C)$$

Logaritmen til feilen som en funksjon av logaritmen til h er en rett linje med stigningstall p . Dette illustreres tydelig ved bruk av et loglog-plot, vist under.

Vi har også tegnet inn en referanse-linje h^p , hvis logaritme er $p \log(h)$, dvs. første leddet uttrykket for $\log|e(h)|$. Hvis teori og praksis stemmer overens skal grafen som viser målt feil ligge parallellt med referanselinjen.

I koden under har vi brukt foroverdifferanser, men prøv med bakoverdifferanser og sentraldifferansen også.

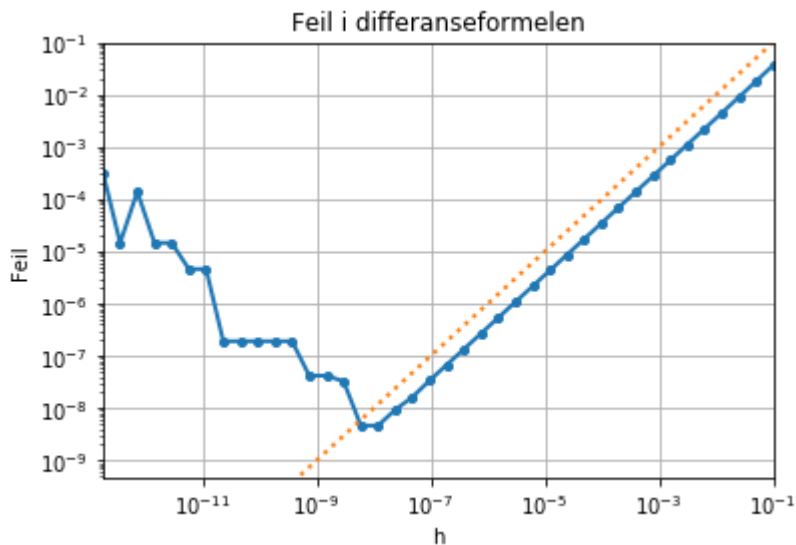
In [4]:

```
x = pi/4;
df_eksakt = cos(x)
h = 0.1
f = sin

Nh = 40          # Antall ganger h halveres.
feil = zeros(Nh) # Array for å lagre feil
steglengder = zeros(Nh) # Array for å lagre h
metode = diff_forover # Velg differanseformel
p = 1             # Metodens orden
for n in range(Nh):
    feil[n] = df_eksakt - metode(f, x, h)
    steglengder[n] = h
    h = 0.5*h

loglog(steglengder, abs(feil), '-.-'); # Feil-graf
loglog(steglengder, steglengder**p, ':') # Referanselinje
xlabel('h')
title('Feil i differanseformelen')
ylabel('Feil')

# Avgrens plottet på en passende måte (kan ignoreres)
y_min = 0.1*min(abs(feil));
axis([steglengder[-1],steglengder[0],y_min, 0.1]);
```



Forklaring:

Reelle tall som lagres i en datamaskin har en avrundingsfeil, se f.eks. [denne wikipediasiden](https://en.wikipedia.org/wiki/Floating-point_arithmetic) (https://en.wikipedia.org/wiki/Floating-point_arithmetic) for detaljer.

La $\hat{y}$ være tallet som representeres i datamaskinen, og y den eksakte verdien. Den relative feilen ε er gitt ved

$$\frac{\hat{y} - y}{y} = \varepsilon \quad \Rightarrow \quad \hat{y} = y \cdot (1 + \varepsilon)$$

og denne er begrenset av $|\varepsilon| < \varepsilon_{mach}$, der ε_{mach} er maskinkonstanten. Denne avhenger av hvordan reelle tall representeres. I vårt tilfelle vil $\varepsilon_{mach} \approx 2.2 \cdot 10^{-16}$ (IEEE, dobbel presisjon). Utover denne begrensningen oppfører ε seg som et tilfeldig tall, og den kan være både positiv og negativ.

Så hva skjer når avrundingsfeil introduseres i en differanseformel, f.eks. foroverdifferansen: La $f_0 = f(x)$, $f_1 = f(x + h)$ og la $\hat{f}_0$ og $\hat{f}_1$ være maskinrepresentasjonen av disse tallene. Det som faktisk beregnes er dermed

$$Df = \frac{\hat{f}_1 - \hat{f}_0}{\hat{h}}.$$

som også kan skrives som

$$\begin{aligned} Df &= \frac{f_1 \cdot (1 + \varepsilon_1) - f_0 \cdot (1 + \varepsilon_2)}{h \cdot (1 + \varepsilon_h)} \\ &\approx \frac{f_1 - f_0 + f_1 \varepsilon_1 - f_0 \varepsilon_2}{h \cdot (1 + \varepsilon_h)} \end{aligned}$$

Nå vet vi at $f_1 - f_0 = f'(x)h + \frac{h^2}{2}f''(\xi)$. Vi kan også anta at $h \approx \hat{h}$, det er ikke ε_h som skaper problemet her. Differanseformelen blir da:

$$Df \approx f'(x) + \frac{h}{2}f''(\xi) + \frac{f_1 \varepsilon_1 - f_0 \varepsilon_2}{h}$$

Vi har altså to bidrag til feilen, en numerisk avbruddsfeil som dominerer feilen for relativt store h , og en feil som skyldes avrundingsfeil, og som vil dominere for h svært liten. Det siste leddet, som altså skyldes avrundingsfeil, er $\sim 1/h$, det blir større (men ganske vilkårlig) når $h \rightarrow 0$.

Minst feil får vi når disse to leddene er like. Antar vi at både f og f'' er av størrelsesorden 1, og at vi ser på størst mulig avrundingsfeil ($-\varepsilon_1 \approx \varepsilon_2 \approx \pm \varepsilon_{mach}$) ser vi at dette skjer omtrent når

$$\frac{h}{2} = \frac{2}{h} \varepsilon_{mach} \quad \Rightarrow \quad h \approx 2\sqrt{\varepsilon_{mach}}$$

og den minste feilen vi kan oppnå er $\approx \sqrt{\varepsilon_{mach}}$. For $\varepsilon_{mach} = 2.2 \cdot 10^{-16}$ kan vi altså regne med at vi oppnår størst nøyaktighet for $h \approx 10^{-8}$ (grovt regnet). Det er også det som observeres i eksempelet over.

Kommentar:

Dette er et eksempel på noe av det verste du kan gjøre med tanke forplantning av avrundingsfeil: Å ta differansen mellom to nesten like tall og dele på et annet lite tall.

Den generaliserte middelverdisetningen

La $f \in C[a, b]$. Gitt k noder x_i i $[a, b]$ og k positive vekter $w_i > 0, i = 1, \dots, k$. Da fins en $\eta \in [\min x_i, \max x_i]$ slik at

$$\sum_{i=1}^n w_i f(x_i) = f(\eta) \sum_{i=1}^k w_i.$$