Andregradslikningen

$$ax^2 + bx + c = 0$$

kan vi løse med formelen

$$x = \frac{b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Men i mange anvendelser dukker det opp likninger ikke kan løses analytisk. Et klassisk eksempel er $x = \cos x$. I disse tilfellene må vi velge blant forskjellige numeriske løsere istedet. En løsning r av en likning kalles gjerne en *rot*.

Halveringsmetoden

Vi søker r slik at

$$f(r) = 0.$$

Dersom f er kontinuerlig og bytter fortegn i r, kan vi benytte denne informasjonen til å finne en approksimasjon x_n til r.

La oss anta at vi kjenner a og b slik at $r \in [a, b]$ er den eneste roten til f i $[a, b]$. Vi definerer nå $x_1 = a$, $x_2 = b$, og $x_3 = \frac{a+b}{2}$. Fortegnet til $f(x_3)$ vil fortelle oss om $x_3 < r$ eller $x_3 > r$, og følgelig vil vi kunne si med sikkerhet om $r \in [x_1, x_3]$ eller $r \in [x_3, x_2]$.

Intervallene $[x_1, x_3]$ og $[x_3, x_2]$ er akkurat halvparten så lange som $[x_1, x_2]$, så etter å ha utført denne prosessen, har vi en mer nøyaktig ide om hvor r befinner seg. Vi setter så $x_2 = x_3$ eller $x_1 = x_3$, alt etter fortegnet til $f(x_3)$, og repeterer prosedyren. Dersom vi gjør dette n ganger, ender vi opp med et intervall med lengde

$$\frac{b-a}{2^n},$$

og hvis vi til slutt setter $r_n = \frac{x_1+x_2}{2}$, vet vi at

$$|r - x_n| \leq \frac{b-a}{2^{n+1}}.$$

Teorem 6.1. *Dersom f er en kontinuerlig funksjon med en rot i intervallet (a, b), produserer n steg med halveringsmetoden et estimat som tilfredsstiller*

$$|r - x_n| \leq \frac{b-a}{2^{n+1}}.$$

Fikspunktmetoden

Anta at vi har en likning på formen

$$r = g(r).$$

Iterasjonen

$$x_{n+1} = g(x_n).$$

vil konvergere mot r under noen omstendigheter. Det kommer litt an på g. Vi begynner med et eksempel.

Eksempel 6.2. Vi løser polynomlikningen

$$x^3 + x^2 - 3x - 3 = 0.$$

Dette polynomet kan spaltes i

$$x^3 + x^2 - 3x - 3 = (x+1)(x-\sqrt{3})(x+\sqrt{3}),$$

og vi skal se hvordan fikspunkmetoden leter etter de forskjellige løsningene. Likningen kan skrives om til $x = g(x)$ på flere måter, men vi skal begynne med å skrive

$$x = \frac{1}{3}\left(x^3 + x^2 - 3\right),$$

slik at

$$g(x) = \frac{1}{3}\left(x^3 + x^2 - 3\right),$$

og

$$x_{n+1} = \frac{1}{3}\left(x_n^3 + x_n^2 - 3\right).$$

Vi prøver å finne løsningen $r = \sqrt{3} \approx 1.732050807568877$, og starter derfor en kjøring i $x_0 = 1.5$. Vi får:

x_0	1.500000000000000
x_5	-0.995705356719772
x_{10}	-0.999982551541273
x_{15}	-0.999999928199386
x_{20}	-0.999999999704524
x_{25}	-0.999999999998784
x_{30}	-0.999999999999995
x_{35}	-1.000000000000000

Også nå ønsker metoden heller å finne $r = -1$. △

Eksemplet over er kjørt med en kodesnutt som ser slik ut:

```
import numpy as np
import time

#initialgjetning
x=-1.5

#denne koden er basert paa at vi overskriver ...
    en enkelt variabel istedet for aa lagre ...
    hele rekken av iterasjoner. derfor ...
    trenger vi en ekstra variabel for aa ...
    holde styr paa avstanden mellom ...
    iterasjonene.
y=x-1

#hvor tett skal det vaere mellom ...
    iterasjonene foer vi er fornoyde?
tol=10**(-16)

#fikspunkt for x-cos x = 0
while np.abs(y-x) > tol: # vi holder paa saa ...
    lenge avstanden mellom iterasjonene er ...
    storre en tol
   y=x # y er naa forrige iterasjon
   x=(x**3+x**2-3)/3 #dette er ...
       fikspunktiterasjonen
   time.sleep(.5) #en liten pause, slik at ...
       vi skal faa tid til aa nyte synet av ...
       den nye iterasjonen
   print x, np.abs(x-y) #som printes her
```

Fikspunktmetoden i forrige eksempel viste en sterk preferanse på hvilken rot den hadde lyst til å finne. Men man kan skrive om likningen til $x = g(x)$ på mange måter.

Eksempel 6.3. Vi skriver nå om likningen

$$x^3 + x^2 - 3x - 3 = 0.$$

til

$$x = \frac{3 + 3x - x^2}{x^2}.$$

Starter vi i $x_0 = -1.5$, får vi

x_0	-1.500000000000000
x_5	-1.732004423011461
x_{10}	-1.732050803458349
x_{15}	-1.732050807568513
x_{20}	-1.732050807568878
x_{25}	-1.732050807568877

Denne fikspunktmetoden klarte fint å finne roten $r = -\sqrt{3}$. △

Som vi ser av de to foregående eksemplene, kan fikspunktmetoden konvergere mot forskjellige røtter avhengig av hvordan vi skriver om likningen. Den kan også ikke konvergere i det hele tatt.

Eksempel 6.4. Vi prøver igjen

$$x = \frac{3 + 3x - x^2}{x^2}.$$

Starter vi i $x_0 = 1.5$, i håp om å finne $r = \sqrt{3}$, får vi

x_0	-1.500000000000000
x_1	2.333333333333333
x_2	0.836734693877551
x_3	6.870315288518744
x_4	-0.499781154362809
x_5	5.007884193672099
x_6	-0.281322161800267
x_7	26.242541136990940
x_8	-0.881325585764740
x_9	-0.541641177723142
x_{10}	3.687092259260734

Denne fikspunktmetoden klarte ikke å finne noe som helst når vi startet i $x_0 = 1.5$. △

For å skjønne hva som skjer, kan vi taylorutvikler g om r:

$$g(x_n) = g(r) + g'(r)(x_n - r) + \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(r)}{3!}(x_n - r)^3 + ...$$

og skrive

$$\begin{aligned} x_{n+1} - r &= g(x_n) - g(r) \\ &= g'(r)(x_n - r) + \frac{g''(r)}{2}(x_n - r)^2 + \dots \end{aligned}$$

og da har vi en likning som sier noe om størrelsen på $x_{n+1} - r$ som en funksjon av $x_n - r$, altså hvor mye feilen minker fra iterasjon til iterasjon. For fikspunktiterasjonen ser vi at det blir best konvergens om g' er så liten som mulig. Vi tar med et konvergensteorem.

Teorem 6.5. *La g være en kontinuerlig deriverbar funksjon. Dersom både $a < g(x) < b$ og $|g'(x)| \le L < 1$ på på $[a, b]$, finnes et entydig punkt r slik at*

$$r = g(r).$$

Fikspunktmetoden

$$x_{n+1} = g(x_n)$$

konvergerer mot r dersom $x_0 \in [a, b]$.

Eksempel 6.6. I eksemplene over er

$$g(x) = \frac{3 + 3x - x^2}{x^2}.$$

og

$$g(x) = \frac{x^3 + x^2 - 3}{3}.$$

Hvis du deriverer disse og evaluerer i røttene til polynomet $x^3 + x^2 - 3x - 3$, vil du se et tydelig mønster. Fikspunktmetoden greier ikke finne r dersom $|g'(r)| > 1$. △

Til slutt kan nevnes at fikspunktmetoden også kan brukes på systemer på formen

$$\mathbf{x} = \mathbf{G}(\mathbf{x})$$

der $\mathbf{x} \in \mathbb{R}^n$ og $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^n$. Iterasjonen er

$$\mathbf{x_{n+1}} = \mathbf{G}(\mathbf{x_n}).$$

Å analysere konvergensen til denne iterasjonen blir for hardt for oss, men vi skal få bruk for selve iterasjonen i kapitlet om numeriske metoder for ordinære differensiallikninger.

Newtons metode

Anta at likningen er på formen

$$f(r) = 0.$$

Prosedyren for Newtons metode er som følger: slå tangenten til funksjonen f i iterasjonen x_n. Punktet der tangenten skjærer x-aksen er den nye iterasjonen x_n. Hvis vi setter opp likningen for tangenten

$$y - f(x_n) = f'(x_n)(x - x_n),$$

og sier at iterasjonen x_{n+1} er den x-verdien slik at $y = 0$:

$$-f(x_n) = f'(x_n)(x_{n+1} - x_n),$$

blir iterasjonen

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Eksempel 6.7. Vi søker løsningene til

$$x^4 - 10x^3 + 35x^2 - 50x + 24 = 0$$

Som Ingrid Espelid Hovig har vi jukset litt, og valgt et polynom som faktoriseres pent:

$$x^4-10x^3+35x^2-50x+24 = (x-1)(x-2)(x-3)(x-4)$$

Newtons metode blir:

$$x_{n+1} = x_n - \frac{x_n^4 - 10x_n^3 + 35x_n^2 - 50x_n + 24}{4x_n^3 - 30x_n^2 + 70x_n - 50}$$

La oss lete etter $r = 1$. Vi starter i $x_0 = 0.5$, og får:

x_0	0.500000000000000
x_1	0.798295454545455
x_2	0.950817599863883
x_3	0.996063283034122
x_4	0.999971872651984
x_5	0.999999998549667
x_6	1.000000000000000

Konvergerer fort dette her. △

Som du ser i eksemplet over, dobles antall korrekte desimaler etter hver iterasjon. For å analysere konvergensen til Newtons metode må vi se den som en fikspunktiterasjon, der

$$g(x) = x - \frac{f(x)}{f'(x)}.$$

Vi beregner

$$g'(x) = 1 - \frac{(f'(x))^2 - f(x)f''(x)}{(f'(x))^2} = \frac{f(x)f''(x)}{(f'(x))^2}$$

Husk at $f(r) = 0$. Dersom $f'(r) \neq 0$, får vi

$$g'(r) = \frac{f(r)f''(r)}{(f'(r))^2} = 0,$$

slik at

$$\begin{aligned} x_{n+1} - r &= g(x_n) - g(r) \\ &= \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(r)}{3!}(x_n - r)^3 + ... \end{aligned}$$

Dette kalles kvadratisk konvergens - feilen etter $n+1$ iterasjoner er proporsjonal med kvadratet av feilen etter n iterasjoner. Da konvergerer det fort.

Eksempel 6.8. Nå prøver vi å finne løsningene til

$$x^4 - 9x^3 + 27x^2 - 31x + 12 = 0.$$

Nok en gang er det et lettfaktorisert polynom:

$$x^4 - 9x^3 + 27x^2 - 31x + 12 = (x-1)^2(x-3)(x-4)$$

Newtons metode blir:

$$x_{n+1} = x_n - \frac{x_n^4 - 9x_n^3 + 27x_n^2 - 31x_n + 12}{4x_n^3 - 27x_n^2 + 54x_n - 31}$$

Nok en gang leter vi etter $r = 1$, ved å starte i $x_0 = 0.5$:

x_0	0.500000000000000
x_1	0.713414634146341
x_2	0.842942878437971
x_3	0.916937117337936
x_4	0.957125910632705
x_5	0.978193460613942
x_6	0.988999465124112
x_7	0.994474755305802
x_8	0.997231047313292
x_9	0.998613930094898
x_{10}	0.999306565270834

Det ser ut til å konvergere, men mye saktere enn i sted. Hva skjedde? △

I eksemplet over er $f'(1) = 0$. Dette betyr at

$$g'(1) = -\frac{f(1)f''(1)}{(f'(1))^2}$$

ikke er bestemt, og grensen

$$\lim_{x\to 1} g'(x)$$

trenger ikke være null. Eksemplet demonstrerer tydelig at den kvadratiske konvergensen ikke kan garanteres dersom $f'(r) = 0$.

Sekantmetoden

Denne ligner på Newton, men vi slår sekanten til f gjennom iterasjonene x_n og x_{i-1}, istedet for tangenten i x_i. Punktet der sekanten skjærer x-aksen er den nye iterasjonen x_n. Iterasjonen blir

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{i-1}}{f(x_n) - f(x_{i-1})}.$$

Merk at her må man ha to initialgjetninger x_1 og x_0, siden man trenger to punkter for å slå en sekant.

Når det gjelder sekantmetoden, kan det vises at

$$x_{n+1} - r = C(x_n - r)^{1.62},$$

men det er et svineri, du må taylorutvikle noe ut av det hinsidige. Konvergens kalles superlineær, altså kjappere enn fikspunkt, men treigere enn Newton.

Newtons metode i to dimensjoner

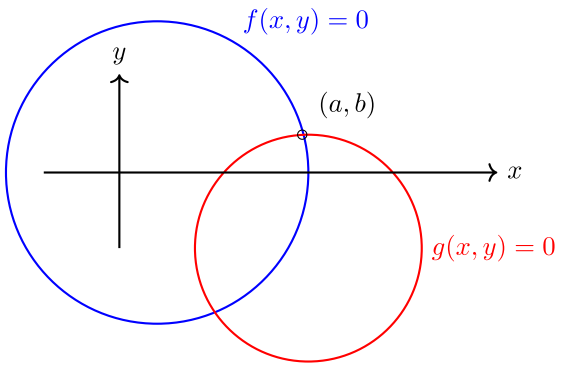
Anta at vi har to funksjoner $f(x, y)$ og $g(x, y)$. Vi leter etter et punkt (a, b) slik at både

$$f(a, b) = 0 \quad \text{og} \quad g(a, b) = 0.$$

likningene

$$f(x, y) = 0 \quad \text{og} \quad g(x, y) = 0$$

angir nivåkurver for funksjonene f og g. Punktet (a, b) må ligge på skjæringspunktet mellom disse.



Anta at du har en iterasjon (x_n, y_n). Vi setter opp tangentplanene til f og g i (x_n, y_n)

$$z - f(x_n, y_n) = f_x(x_n, y_n)(x - x_n) + f_y(x_n, y_n)(y - y_n)$$

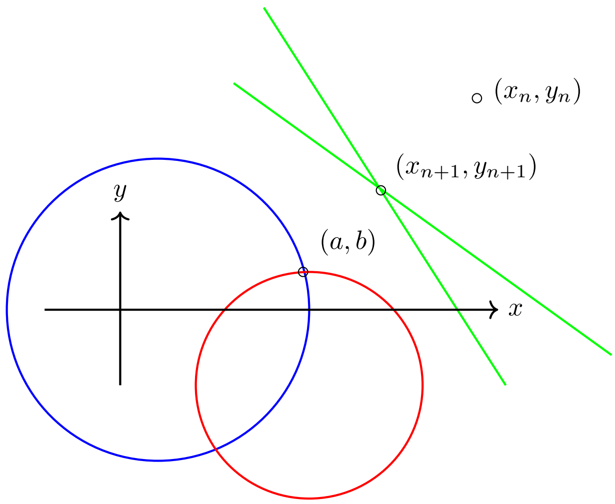
$$z - g(x_n, y_n) = g_x(x_n, y_n)(x - x_n) + g_y(x_n, y_n)(y - y_n).$$

Hvis vi krever at $z = 0$, får vi likninger for skjæringslinjene mellom disse tangentplanene og (x, y)-planet

$$-f(x_n, y_n) = f_x(x_n,y_n)(x - x_n) + f_y(x_n, y_n)(y - y_n)$$

$$-g(x_n, y_n) = g_x(x_n,y_n)(x - x_n) + g_y(x_n, y_n)(y - y_n).$$

Iterasjonen (x_{n+1}, y_{n+1}) defineres som skjæringen mellom disse linjene.



Altså er

$$-f(x_n, y_n) = f_x(x_n,y_n)(x_{n+1} - x_n) + f_y(x_n, y_n)(y_{n+1} - y_n)$$

$$-g(x_n, y_n) = g_x(x_n,y_n)(x_{n+1} - x_n) + g_y(x_n, y_n)(y_{n+1} - y_n),$$

et lineært likningssystem som definerer (x_{n+1}, y_{n+1}). Matrisen er

$$\begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix}$$

og vi skriver

$$-\begin{pmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{pmatrix} = \begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix} \begin{pmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{pmatrix}.$$

Nå kan vi gange med

$$\begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix}^{-1}$$

fra venstre, legge til (x_n, y_n) på begge sider, og få Newtons metode for systemer

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix}^{-1} \begin{pmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{pmatrix}.$$

Jeg gadd ikke skrive ut hva

$$\begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix}^{-1}$$

er, men du kan regne det ut ved å huske fra lineæralgebraen at

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Eksempel 6.9. Vi leter etter løsninger til systemet

$$x^2 + y^2 = 1$$

$$\frac{x^2}{4} + 2y^2 = 1$$

Som du husker fra M2 er den første likningen for enhetssirkelen, mens den andre er likningen for en ellipse med halvakser 2 og $\frac{1}{2}$. Trekker man to ganger den første fra den andre, kan man regne ut at skjæringspunktet mellom disse kurvene i første kvadrant er

$$\left(\frac{2}{\sqrt{7}}, \sqrt{\frac{3}{7}}\right) \approx (0.755928946018454, 0.654653670707977).$$

La oss se om Newtons metode finner dette punktet. Siden

$$f(x, y) = x^2 + y^2 - 1$$

og

$$g(x, y) = \frac{x^2}{4} + 2y^2 - 1$$

blir

$$\begin{pmatrix} f_x(x_n, y_n) & f_y(x_n, y_n) \\ g_x(x_n, y_n) & g_y(x_n, y_n) \end{pmatrix} = \begin{pmatrix} 2x & 2y \\ \frac{x}{2} & 4y \end{pmatrix}$$

som har invers

$$\frac{1}{7xy} \begin{pmatrix} 4y & -2y \\ -\frac{x}{2} & 2x \end{pmatrix}.$$

Metoden blir

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \frac{1}{7xy} \begin{pmatrix} 4y & -2y \\ -\frac{x}{2} & 2x \end{pmatrix} \begin{pmatrix} f(x_n, y_n) \\ g(x_n, y_n) \end{pmatrix},$$

og starter vi i $(1, 1)$, får vi:

n	x_n	y_n
1	1.000000000000000	1.000000000000000
2	0.785714285714286	0.714285714285714
3	0.756493506493507	0.657142857142857
4	0.755929156680230	0.654658385093168
5	0.755928946018484	0.654653670724952
6	0.755928946018454	0.654653670707977
7	0.755928946018455	0.654653670707977

Maskinpresisjon etter syv iterasjoner. Bra greier det her. $\triangle$