TMA4135 Matematikk 4D

Forelesningsnotater

Numerikk

Disse forelesningsnotatene er ikke noen erstatning for læreboken, for den er en murstein, altså tar den flere tiår, en haug med slaver, samt et stort forlag, å skrive. Feil sniker seg inn - finner du noen, send gjerne epost til morten.nome@gmail.com.

Ligningsløsere

I anvendelser møter man ligninger som ikke lar seg løse analytisk. Et klassisk eksempel er $x = \cos x$.

Fikspunktiterasjonen

Anta at vi har en ligning på formen

$$r = g(r).$$

Iterasjonen

$$x_{i+1} = g(x_i). \tag{1}$$

vil konvergere mot r dersom $r, x_0 \in (a, b)$ og $g'(x) < 1$ på (a, b).

Newtons metode

Anta at ligningen er på formen

$$f(r) = 0.$$

Prosedyren for Newtons metode er som følger: slå tangenten til funksjonen f i iterasjonen x_i. Punktet der tangenten skjærer x-aksen er den nye iterasjonen x_{i+1}. Hvis vi setter opp ligningen for tangenten

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

og sier at iterasjonen x_{i+1} er de x-verdien slik at $y = 0$

$$-f(x_i) = f'(x_i)(x_{i+1} - x_i).$$

blir iterasjonen

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}.$$

Sekantmetoden

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

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

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

Newtons metode i to dimensjoner

Anta at vi har to funksjoner $f_1(x,y)$ og $f_2(x,y)$. Vi leter etter et punkt (a,b) slik at både

$$f_1(a,b) = 0$$

og

$$f_2(a,b) = 0.$$

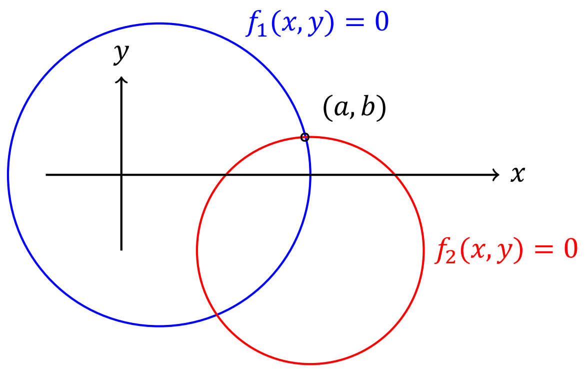
Ligningene

$$f_1(x,y) = 0$$

og

$$f_2(x,y) = 0$$

angir nivåkurver for funksjonene f_1 og f_2. Punktet (a,b) må ligge på skjæringspunktet mellom disse.



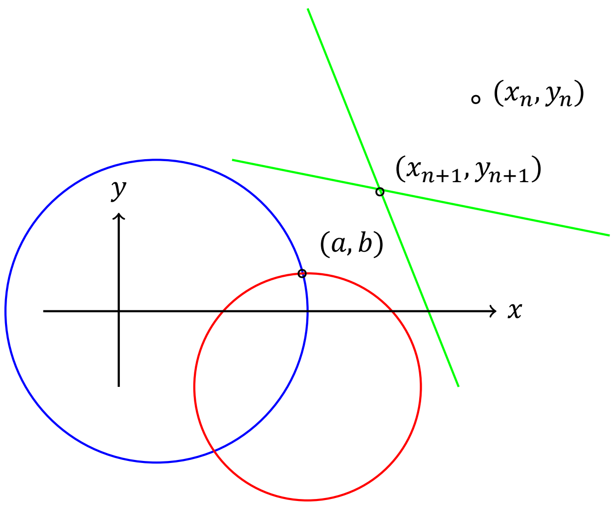
Anta at du har en iterasjon (x_n, y_n). Vi setter opp tangentplanene til f_1 og f_2 i (x_n, y_n)

$$z - f_1(x_n, y_n) = \frac{\partial f_1}{\partial x}(x_n, y_n)(x - x_n) + \frac{\partial f_1}{\partial y}(x_n, y_n)(y - y_n)$$
$$z - f_2(x_n, y_n) = \frac{\partial f_2}{\partial x}(x_n, y_n)(x - x_n) + \frac{\partial f_2}{\partial y}(x_n, y_n)(y - y_n).$$

Hvis vi krever at $z = 0$, får vi ligninger for der disse tangentlinjene skjærer (x, y)-planet

$$-f_1(x_n, y_n) = \frac{\partial f_1}{\partial x}(x_n, y_n)(x - x_n) + \frac{\partial f_1}{\partial y}(x_n, y_n)(y - y_n)$$
$$-f_2(x_n, y_n) = \frac{\partial f_2}{\partial x}(x_n, y_n)(x - x_n) + \frac{\partial f_2}{\partial y}(x_n, y_n)(y - y_n).$$

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



Altså er

$$-f_1(x_n, y_n) = \frac{\partial f_1}{\partial x}(x_n, y_n)(x_{n+1} - x_n) + \frac{\partial f_1}{\partial y}(x_n, y_n)(y_{n+1} - y_n)$$
$$-f_2(x_n, y_n) = \frac{\partial f_2}{\partial x}(x_n, y_n)(x_{n+1} - x_n) + \frac{\partial f_2}{\partial y}(x_n, y_n)(y_{n+1} - y_n).$$

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

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}(x_n, y_n) & \frac{\partial f_1}{\partial y}(x_n, y_n) \\ \frac{\partial f_2}{\partial x}(x_n, y_n) & \frac{\partial f_2}{\partial y}(x_n, y_n) \end{pmatrix}$$

og vi skriver

$$-\begin{pmatrix} f_1(x_n, y_n) \\ \\ f_2(x_n, y_n) \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial x}(x_n, y_n) & \frac{\partial f_1}{\partial y}(x_n, y_n) \\ \\ \frac{\partial f_2}{\partial x}(x_n, y_n) & \frac{\partial f_2}{\partial y}(x_n, y_n) \end{pmatrix} \begin{pmatrix} x_{n+1} - x_n \\ \\ y_{n+1} - y_n \end{pmatrix}.$$

Nå kan vi gange med

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}(x_n, y_n) & \frac{\partial f_1}{\partial y}(x_n, y_n) \\ \\ \frac{\partial f_2}{\partial x}(x_n, y_n) & \frac{\partial f_2}{\partial y}(x_n, y_n) \end{pmatrix}^{-1}$$

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

$$\begin{pmatrix} x_{n+1} \\ \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ \\ y_n \end{pmatrix} - \begin{pmatrix} \frac{\partial f_1}{\partial x}(x_n, y_n) & \frac{\partial f_1}{\partial y}(x_n, y_n) \\ \\ \frac{\partial f_2}{\partial x}(x_n, y_n) & \frac{\partial f_2}{\partial y}(x_n, y_n) \end{pmatrix}^{-1} \begin{pmatrix} f_1(x_n, y_n) \\ \\ f_2(x_n, y_n) \end{pmatrix}.$$

Jeg gadd ikke skrive ut hva

$$\begin{pmatrix} \frac{\partial f_1}{\partial x}(x_n, y_n) & \frac{\partial f_1}{\partial y}(x_n, y_n) \\ \\ \frac{\partial f_2}{\partial x}(x_n, y_n) & \frac{\partial f_2}{\partial y}(x_n, y_n) \end{pmatrix}^{-1}$$

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

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

Litt om konvergens

Vi bruker taylorutvikling for å analysere hvor fort en ligningsløser konvergerer mot løsningen. La oss begynne med fikspunktiterasjonen. Vi taylorutvikler g om r, og får

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

Nå kan vi skrive

$$x_{n+1} - r = g(x_n) - g(r) = g'(r)(x_n - r) + \frac{g''(r)}{2}(x_n - r)^2 + \frac{g'''(r)}{3!}(x_n - r)^3 + \dots$$

og da har vi en ligning 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 flat som mulig. Konvergensen kalles for øvrig lineær, siden $x_{n+1} - r$ er proporsjonal med $x_n - r$.

For å analysere Newtons metode

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},$$

må vi se på det som en fikspunktiterasjon der

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

Vi beregner

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

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

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

slik at

$$x_{n+1} - r = g(x_n) - g(r) = \frac{g''(r)}{2}(x_n - r)^2 + O((x_n - r)^3).$$

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

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.

Polynominterpolasjon

Vi tenker oss at vi har $n+1$ punkter på intervallet $[a, b]$, og en funksjon $f : [a, b] \to \mathbb{R}$. Punktene kaller vi x_i der $0 \leq i \leq n$, med $x_0 = a$, $x_n = b$. For korthets skyld skriver vi $f(x_i) = f_i$. Oppgaven er å finne et polynom av grad n som reiser gjennom punktene (x_i, f_i).

Lagranges interpolasjon

For hvert punkt x_i, definerer vi et polynom $L_i(x)$ av orden n:

$$L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{(x - x_k)}{(x_i - x_k)}.$$

Polynomet $L_i(x)$ har egenskapen

$$L_i(x_k) = \begin{cases} 1 & \text{for } i = k \\ 0 & \text{for } i \neq k \end{cases}.$$

Nå er det lett å sette opp et interpolerende polynom:

$$p_n(x) = \sum_{i=0}^{n} f_i L_i(x).$$

Merk at $p_n(x_k) = f(x_k)$ for alle k. Konstruksjonen av Lagranges interpolasjon viser at det for en tabell med $n+1$ punkter eksisterer et interpolasjonspolynom av grad n; vi har jo nettopp konstruert det. Hvis vi antar at det finnes to forskjellige polynomer p_n og q_n av grad n som interpolerer den samme tabellen, og evaluerer differansen $p_n - q_n$ i punktene x_i, ser vi at

$$p_n(x_i) - q_n(x_i) = 0 \qquad 0 \leq i \leq n.$$

Men polynomet $p - q$ har grad n, og kan følgelig ikke ha $n+1$ nullpunkter, altså må $p = q$. Dette betyr at interpolasjonspolynomet er unikt.

Newtons interpolasjon

For å konstruere Newtons interpolasjon trenger vi først å beregne noe som kalles de dividerte differansene. De defineres rekursivt slik:

$$f[x_0] = f(x_0)$$

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$f[x_n, \dots, x_0] = \frac{f[x_n, \dots, x_1] - f[x_{n-1}, \dots, x_0]}{x_n - x_0}$$

Nå kan vi sette opp Newtons interpolasjon:

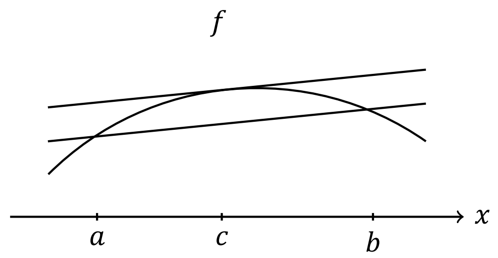
$$p_n(x) = f_0 + \sum_{i=1}^{n} f[x_i, \dots, x_0] \prod_{k=0}^{i-1} (x - x_k).$$

Merk at Lagranges og Newtons interpolasjon er bare to forskjellige formler for å sette opp det samme polynomet, siden interpolasjonspolynomet er unikt.

For de dividerte differansene kan man sette opp en generalisering av middelverdisatsen. Middelverdisatsen sier at

$$f'(c) = \frac{f(b) - f(a)}{b - a} = f[a, b]$$

for en funksjon som er deriverbar på $[a, b]$.



Den tilsvarende satsen for dividerte differenser sier at dersom f er $n + 1$ ganger deriverbar, og alle punktene x_i er forskjellige, er

$$f[x_n, \dots, x_0] = \frac{f^n(s)}{n!}$$

for en eller annen s i intervallet $[a, b]$. Beviset er ikke så vrient. Siden p_n interpolerer f, funksjonen $g = f - p_n$ ha $n + 1$ nullpunkt på intervallet $[a, b]$. Gjentatt anvendelse av middelverdisatsen forteller oss at g' har n nullpunkt, at g'' har $n - 1$ nullpunkt, og videre at g^{n+1} har ett nullpunkt på $[a, b]$. Vi kaller dette s. Siden

$$\frac{d^n}{dx^n} p_n(x) = n!\, f[x_n, \dots, x_0],$$

må

$$f^n(s) = n!\, f[x_n, \dots, x_0].$$

Interpolasjonsfeilen

Vi skriver opp Newtons interpolasjonspolynom

$$p_n(x) = f_0 + \sum_{i=1}^{n-1} f[x_i, \dots, x_0] \prod_{k=0}^{i-1} (x - x_k) + f[x_n, \dots, x_0] \prod_{k=0}^{n-1} (x - x_k).$$

Jeg har skrevet ut det siste leddet kun av pedagogiske hensyn. Nå bytter vi ut x_n med x i uttrykket over (tenk på x som et nytt interpolasjonspunkt), og får

$$f(x) = f_0 + \sum_{i=1}^{n-1} f[x_i, \ldots, x_0] \prod_{k=0}^{i-1}(x - x_k) + f[x, x_{n-1} \ldots, x_0] \prod_{k=0}^{n-1}(x - x_k),$$

slik at

$$f(x) - p_n(x) = (f[x, x_{n-1} \ldots, x_0] - f[x_n, \ldots, x_0]) \prod_{k=0}^{n-1}(x - x_k) = f[x, x_n, \ldots, x_0] \prod_{k=0}^{n}(x - x_k) = \frac{f^{n+1}(s)}{(n+1)!} \prod_{k=0}^{n}(x - x_k),$$

for en eller annen $s \in [a, b]$. Merk at s avhenger av x.

Punktfordeling

Det er langt fra likegyldig hvilke punkter $\{x_i\}$ som brukes i interpolasjonen. Ekvidistante punkter er håpløst. Chebyshevpunktene $x_i = \cos \frac{j\pi}{n}$ for $0 \le i \le n$ er best. Ingen over, noen få ved siden. Man finner gode punkter for interpolasjon ved å plassere dem slik at de minimerer utslaget til polynomet

$$\prod_{k=0}^{n}(x - x_k)$$

i interpolasjonsfeilen.