Kapitel 4. Iterativ lösning av ekvationer

Vi skall nu undersöka, har man löser numeriskt ekvationer av formen $f(x) = 0$. Dylika ekvationer kallas också olinjära, eftersom funktionen oftast har ett olinjärt beroende av x.

Olinjära ekvationer behöver inte ha entydiga lösningar, eller ens en lösning i sluten form. Att femtegradsekvationen inte kan lösas i sluten form visades av *Niels Henrik Abel* år 1824 i en liten "Memoire" [1] på sex sidor, som negligerades av den tidens stora matematiker (Gauss och Cauchy m.fl.).

Uppenbarligen kan olinjära ekvationer lösas endast approximativt. Vad menar vi då med en approximativ lösning? Antag, att x^* satisfierar $f(x) = 0$. Vi kan då säga, att $\tilde{x}$ är en (approximativ) lösning till $f(x) = 0$ antingen om

$$|f(\tilde{x})| \approx 0 \quad eller \quad |\tilde{x} - x^*| \approx 0.$$

[1] N.H. Abel: *Mémoire sur les équations algébriques ou l'on démontre l'impossibilité de la resolution de l'équation général du cinquième degré*, Œuvres complétes, Vol. I, Christiania, 1881

Det andra av de båda kriterierna förutsätter, att man vet att ekvationen har en lösning. Detta är inte så underligt, ty om funktionen f är kontinuerlig, kan man alltid finna ett intervall där funktionen byter förtecken. Ofta överensstämmer de båda metoderna att definiera en "lösning".

I praktiken använder man vanligen iterativa metoder för att lösa en olinjär ekvation, dvs man beräknar en räcka av approximativa lösningar $x_1, x_2, \ldots, x_n, \ldots$ som *konvergerar* mot lösningen till ekvationen $f(x) = 0$. Sådana beräkningar lämpar sig därför mycket väl för datorer.

Vad innebär konvergensen av en iterativ räcka av approximationer? Matematiskt kan konvergensvillkoret formuleras på följande sätt: En räcka av reella tal a_n sägs konvergera mot ett gränsvärde a (dvs $a_n \longrightarrow a$ då $n \longrightarrow \infty$ eller $\lim_{n\to\infty} a_n = a$) om det mot varje $\epsilon > 0$ svarar ett positivt tal N så beskaffat, att

$$|a_n - a| < \epsilon$$

för varje $n > N$.

Ofta brukar man omskriva villkoret $|a_n - a| < \epsilon$ i formen

$$a - \epsilon < a_n < a + \epsilon$$

4.1. Funktionsiteration

En av de enklaste metoderna att lösa en ekvation $f(x) = 0$ får man om den omskrivs i formen $x = g(x)$, och denna form av ekvationen används för att definiera en iterativ process.

Vi utgår från en uppskattning x_0 till lösningen, och definierar en räcka av successiva approximationer x_n till roten med hjälp av ekvationerna

$$x_n = g(x_{n-1}), \quad n = 1, 2, \ldots$$

Om g är en kontinuerlig funktion (iterationsfunktionen), och denna räcka konvergerar, så kommer approximationerna att närma sig varandra mer och mer. Till sist får man med godtycklig noggrannhet $x_n \approx x_{n-1}$, dvs

$$x_{n-1} \approx g(x_{n-1})$$

så att x_n alltså är en approximativ lösning till ekvationen $x = g(x)$.

Om detta tillämpas på ekvationen $x - \cos x = 0$, som kan omskrivas i formen $x = \cos x$, så får vi med utgångsapproximationen $x_0 = 0.7$ de successiva approximationerna $x_1 = \cos 0.7 \approx 0.7648$, $x_2 = \cos 0.7648 \approx 0.7215$, $x_3 = \cos 0.7215 \approx 0.7508$, ..., $x_9 \approx 0.7402$, $x_{10} \approx 0.7383$.

Med två decimalers noggrannhet kan man alltså anse lösningen vara 0.74. Men detta bevisar ännu ingenting om konvergensen.

Om vi skulle ha skrivit ekvationen i formen $x = \arccos x$ istället, så skulle vi med utgångsvärdet $x = 0.74$ få iterationsräckan 0.7377, 0.7411, 0.7361, 0.7435, 0.7325, 0.7489, 0.7245, 0.7605, 0.7067, 0.7860, 0.6665, 0.8414, 0.5710, 0.9631, 0.2727, 1.2946, som uppenbarligen inte leder någon vart!

Kan vi veta på förhand om en räcka konvergerar, och uppskatta noggrannheten? Som vi skall se, är svaret ja. Man kan studera konvergensen grafiskt genom att upprita funktionerna $y = x$ och $y = g(x)$ i ett (x, y)–koordinatsystem. Punkten x_{n-1} finner man genom att upprita den vertikala linjen $x = x_{n-1}$ tills den når kurvan $y = g(x)$. Den horisontella linjen $y = g(x_{n-1})$ kommer sedan att träffa kurvan $y = x$ i punkten $x = x_n = g(x_{n-1})$. Om $g(x)$ är en (monotont) avtagande funktion, så kommer approximationerna att konvergera via oskillationer mot lösningen. Approximationerna kan också konvergera monotont.

Antag nu, att vi vet att det finns en rot r till ekvationen $x = g(x)$, och låt oss beteckna felen i de olika approximationerna med e_n. Felen kan då uttryckas $e_n = x_n - r$.

Om vi utvecklar g i Taylors serie kring $x = r$, och beaktar att $g(r) = r$, så får vi

$$\begin{aligned} e_{n+1} &= x_{n+1} - r = g(x_n) - g(r) \\ &= \left[g(r) + (x_n - r)g'(r) + \frac{(x_n - r)^2}{2!} g''(r) + \ldots \right] - g(r) \\ &= e_n g'(r) + \frac{e_n^2}{2!} g''(r) + \ldots \end{aligned}$$

Om $|e_n|$ är så litet att vi kan försumma termer av högre ordning, så gäller $e_{n+1} \approx e_n g'(r)$ varav följer att felet kan reduceras om $|g'(r)| < 1$. Tyvärr har vi inte så stor nytta av detta resultat, eftersom vi inte känner roten r.

Men man kan också visa, att ett tillräckligt villkor för att iterationsräckan konvergerar är att $|g'(x)| < 1$ gäller inom hela det betraktade intervallet. Detta följer av följande sats: Antag, att g är differentierbar inom intervallet $[a, b]$, och att 1) $g([a, b]) \subseteq [a, b]$ (dvs $g(x) \in [a, b]\ \forall x \in [a, b]$), och att 2) $|g'(x)| \leq K < 1\ \forall x \in [a, b]$.

I detta fall har ekvationen $x = g(x)$ en entydig lösning inom intervallet $[a, b]$ och den iterativa räcka som definieras av

$$x_0 \in [a, b]; \quad x_n = g(x_{n-1}), n = 1, 2 \ldots$$

kommer att konvergera mot denna lösning.

Iterationsräckorna kan konvergera olika snabbt, beroende på värdet av iterationsfunktionens derivator. Detta följer av Taylorutvecklingen av felet:

$$e_{n+1} = e_n g'(r) + \frac{e_n^2}{2!} g''(r) + \ldots$$

Om $g'(r) \neq 0$, så är $e_{n+1} \propto e_n$, och konvergenshastigheten är linear, men om $g'(r) = 0$ och $g''(r) \neq 0$, så är $e_{n+1} \propto e_n^2$, och konvergenshastigheten är kvadratisk. Mer om detta senare.

4.2. Bisektionsmetoden

Ett av de enklaste sätten att lösa ekvationen $f(x) = 0$ är den s.k. bisektionsmetoden, som baserar sig på *medelvärdessatsen*: Antag, att f är en funktion, som är kontinuerlig över ett intervall $[a, b]$, och att $f(a)f(b) < 0$ (så att funktionen är positiv i den ena ändpunkten av intervallet och negativ i den andra ändpunkten). Då finns det åtminstone en lösning till ekvationen $f(x) = 0$ mellan a och b.

Bisektionsmetoden tillämpas så, att man halverar intervallet $[a, b]$, betecknar mittpunkten med m och upprepar proceduren på det av de två delintervallen $[a, m]$ och $[m, b]$ inom vilket funktionen byter förtecken. Då intervallängden är mindre än ett visst litet tal (toleransen), avslutas räkningen.

Den fullständiga algoritmen kan uttryckas på följande sätt:

Vi antar, att intervallet $[a, b]$ är sådant, att $f(a)f(b) < 0$.

1. Räkningen avslutas då $b - a \leq \epsilon_1$.
2. Låt $m = \frac{1}{2}(a + b)$ vara mittpunkten av intervallet $[a, b]$. Beräkna $f(m)$. Räkningen avslutas då $|f(m)| \leq \epsilon_2$.
3. Låt det nya intervallet vara $[a, m]$, ifall $f(a) \cdot f(m) < 0$, välj i annat fall $[m, b]$, och gå till steg 1.

I denna algoritm undersöks också möjligheten av att funktionsvärdet av en händelse blir 0. Vid varje iteration halveras intervallet. Om medelpunkten m anses som en approximation till roten x^*, så reduceras även felet $|m - x^*|$ med hälften vid varje iteration. Vid flyttalsaritmetik med 32 bitars ord och 24 bitar i mantissan borde man alltså få full noggrannhet efter 24 iterationer. Om felet vid den i:te iterationen betecknas $e_i = m - x^*$, så fås alltså $|e_{i+1}|/|e_i| \approx \frac{1}{2}$. I allmänhet säger man att konvergensraten är r ifall $\lim_{i\to\infty}(|e_{i+1}|/|e_i|^r) = C$, där C är en konstant. I bisektionsmetoden är $r = 1$ (linjär konvergens) och $C = \frac{1}{2}$. Om $r = 2$ talar man om en kvadratisk konvergens. Högre värden av r ger i allmänhet snabbare konvergens, men om C är litet, kan också en linjär rat ge snabb konvergens.

Vi skall se på ett enkelt exempel: $x - \cos x = 0$. Om vi sätter $f(x) = x - \cos x$, så får vi $f(0) = -1 < 0$ samt $f(\pi/2) = \pi/2 > 0$. Eftersom f är kontinuerlig inom intervallet, så måste det innehålla en lösning till ekvationen. Vi sätter alltså $a = 0$ och $b = \pi/2$. Mittpunkten av detta intervall är $m = \pi/4$. Eftersom $f(\pi/4) = \pi/4 - \cos \pi/4 \approx 0.07829 > 0$, så finns lösningen inom intervallet $[0, \pi/4]$, vars mittpunkt är $\pi/8$. Emedan $f(\pi/8) \approx -0.53118 < 0$, så ligger lösningen till ekvationen inom intervallet $[\pi/8, \pi/4]$, vars mittpunkt är $m = 3\pi/16$. Då $f(3\pi/16) \approx -0.24242 < 0$, så finner vi att lösningen måste ligga inom intervallet $[3\pi/16, \pi/4]$. Mittpunkten av detta intervall är $m = 7\pi/32$, där funktionen antar värdet $f(7\pi/32) \approx -0.08579 < 0$, så att lösningen måste ligga inom intervallet $[7\pi/32, \pi/4]$. Proceduren kan upprepas tills vi når en önskad toleransgräns.

Nedan visas ett MATLAB-program, som löser en ekvation enligt bisektionsmetoden:

```
function rot=bisekt(funk,a,b,eps1,eps2)
% Bisektionsmetoden för funk(x)=0
% med toleransen eps1, eps2 inom intervallet [a,b]
% Funktionen funk beräknas i en m-fil
fa = feval(funk,a); fb = feval(funk,b);
if fa*fb>0
   fprintf('samma tecken i a och b!\n')
   return
end
while abs(b-a) > 2*eps1
   m = (a+b)/2;
   fm=feval(funk,m);
   if abs(fm)<=eps2, break;
   end
   if fa*fm<=0, b=m;
   else a=m; end
end
rot=(a+b)/2;
```

4.3. Newtons metod och sekantmetoden

En av de bästa metoderna för att lösa ekvationen $f(x) = 0$ är härrör sig från *Isaac Newton*. Låt oss anta att x_i är en approximation till lösningen x^*. I Newtons metod approximerar man $f(x)$ med dess tangent i punkten x_i. Tangentens nollställe väljs därpå till ny approximation för x^*. Analytiskt kan man beskriva detta på följande sätt: Vi utvecklar $f(x)$ i en Taylors serie kring punkten x_i:

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \dots .$$

Tangentens ekvation finner man av de två första termerna i serien:

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

och genom att sätta $y = 0$ får vi

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

I Newtons metod används således iterationsfunktionen

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

som har derivatan

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

Emedan $f(r) = 0$, då r är den exakta roten, så blir $g'(r) = 0$ som man kunde vänta sig. Vi väntar oss därför att Newtons metod har kvadratisk konvergens (se nedan).

Newton gjorde sin metod känd i *Principia*[2] genom sin lösning av Keplers ekvation $x - e\sin x = M$, där e betecknar excentriciteten, M är medelanomalin (som är känd) och x är den excentriska anomalin (som skall beräknas) för en planetbana. Om vi sätter $f(x) = x - e\sin x - M$, och man känner en approximation x_i för roten, så kan man beräkna en ny approximation ur ekvationen $x_{i+1} = x_i - f(x_i)/f'(x_i) = x_i - (x_i - e\sin x_i - M)/(1 - e\cos x_i)$. Tyvärr fungerar inte Newtons metod särskilt bra för stora värden av excentriciteten, för kometbanor lämpar sig därför bisektionsmetoden bättre.

[2] *Principia Mathematica*, Book I, Prop. XXXI, Probl. XXIII och Scholium, fast beskrivningen där inte är lätt att förstå.

Proceduren diskuterades ingående av *Joseph Raphson* i *Analysis Aequationum Universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nova infinitarum serium doctrina deducta ac demonstrata* år 1690, där Raphson hänvisade till Newton som metodens upptäckare.

Det är ganska lätt att visa att Newtons metod har en kvadratisk konvergens. Om $f(x)$ utvecklas i Taylors serie kring x_i fås

$$f(x) = f(x_i) + f'(x_i)(x - x_i) + \tfrac{1}{2}f''(\xi)(x - x_i)^2,$$

där ξ är en punkt mellan x och x_i. Om $x = x^*$ (den exakta lösningen), fås

$$0 = f(x^*) = f(x_i) + f'(x_i)(x^* - x_i) + \tfrac{1}{2}f''(\xi)(x^* - x_i)^2.$$

Om denna ekvation divideras med $f'(x_i)$ och grupperas om, får man

$$x^* - \left(x_i - \frac{f(x_i)}{f'(x_i)} \right) = \frac{f''(\xi)}{2f'(x_i)}(x^* - x_i)^2.$$

På grund av Newtons iterationsformel blir vänstra membrum av denna ekvation $x^* - x_{i+1}$, och vi kan skriva

$$e_{i+1} = C_i e_i^2,$$

ifall $e_i = x^* - x_i$ och $C_i = f''(\xi)/2f'(x_i)$. Om processen är konvergent, så är

$$\lim_{i \to \infty} \frac{|e_{i+1}|}{|e_i|^2} = C,$$

där $C = |f''(x^*)/2f'(x^*)|$, vsb.

Som ett enkelt exempel på användningen av Newtons metod skall vi visa, hur man kan använda den för att beräkna den positiva kvadratroten av ett reellt tal a. Om vi tillämpar Newtons metod på ekvationen $x^2 - a = 0$ får vi

$$x_{n+1} = x_n - \frac{x_n^2 - a}{2x_n} = \frac{x_n + a/x_n}{2}$$

Denna iterationsräcka konvergerar för varje $x_0 > 0$. Antag t.ex. att $a = 5$ och starta från $x_0 = 2$. Då är $x_1 = (2 + 5/2)/2 = 2.25$, $x_2 \approx 2.236111$, och $x_3 \approx 2.236068$, som är $\sqrt{5}$ med 6 decimaler.

Newtons metod är mycket lätt att programmera t.ex. med MATLAB:

```
funktion rot=newton(funk,df,x,eps1)
% Newtons metod för ekvationen funk(x)=0
% med toleransen eps1<1. Utgångsvärdet är x.
% funk och derivatan df beräknas i m-filer.
ori=x+1;
while abs(x-ori)>eps1
    ori=x;
    x=ori-feval(funk,ori)/feval(df,ori);
end
rot=x;
```

Newtons metod fungerar inte alltid så bra. Den behöver t.ex. inte konvergera. Om $f'(x_i) = 0$ så är den inte väldefinierad, och även om $f'(x_i) \approx 0$ kan det uppstå svårigheter, eftersom den nya approximationen x_{i+1} kan vara en sämre approximation till ekvationens lösning än x_i. Newtons metod kan emellertid förbättras så att sådana problem inte behöver uppträda. Ett annat problem med Newtons metod är att man måste beräkna $f'(x)$, vilket kan vara tidskrävande, eller t.o.m. omöjligt. Ett sätt att kringgå detta problem är att göra en numerisk approximation av derivatan.

Ett exempel på en ekvation som uppvisar detta problem är $f(x) = \arctan(x-1) - 0.5$. Den exakta lösningen till denna ekvation är $\tan 0.5 + 1 \approx 1.5463$. Om vi utgår från $x_0 = 4$ och använder Newtons metod får vi $x_1 \approx -3.5$ och $x_2 \approx 36$. Oskillationerna blir allt vildare, och iterationerna divergerar. Orsaken till divergensen är i detta fall att andra derivatan har ett nollställe nära roten.

Följande allmänna konvergenssats gäller för Newtons metod. Antag, att de två första derivatorna av f existerar inom intervallet $[a, b]$ och att följande villkor gäller:

1. $f(a)f(b) < 0$
2. f' har inga nollställen inom intervallet $[a, b]$
3. f'' ändrar inte förtecken inom intervallet $[a, b]$ samt att
4. $\left|\frac{f(a)}{f'(a)}\right|, \left|\frac{f(b)}{f'(b)}\right| < b - a$

I ett sådant fall finns det en entydig lösning $r \in (a, b)$ till ekvationen $f(x) = 0$, och Newtons metod kommer att konvergera mot r från en godtycklig punkt inom intervallet $[a, b]$.

Ett sätt lösa konvergensproblemet är att använda sekantmetoden, som inte behöver några derivator. I denna metod använder man två tidigare approximationer x_i och x_{i-1}, och bestämmer en rät linje (*sekanten*) genom punkterna $(x_i, f(x_i))$ och $(x_{i-1}, f(x_{i-1}))$. Sekantens nollställe väljs som ny approximation för x^*.

Analytiskt kan man gå till väga på följande sätt. Ekvationen för den räta linje, som passerar genom punkterna $(x_i, f(x_i))$ och $(x_{i-1}, f(x_{i-1}))$ är

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

Genom att välja $y = 0$ och $x = x_{i+1}$ fås därpå

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

som är den formel som används i sekantmetoden. Den motsvarar formeln i Newtons metod om vi använder den approximativa formeln för derivatan

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

När man använder sekantmetoden, behöver man inte beräkna derivator, men det gör även konvergensen långsammare.

Sekantmetoden konvergerar snabbare än bisektionsmetoden ("superlinjärt") med konvergensraten $r = \frac{1}{2}(1+\sqrt{5}) \approx 1.618$ och konstanten $C = |f''(x^*)/2f'(x^*)|^{1/r}$ (något besvärligt att bevisa). Felet avtar rätt snabbt mot noll, men inte lika snabbt som i Newtons metod. Man kan uttrycka konvergensen så, att antalet korrekta decimaler växer med omkring 60 % för varje iteration. Metoden har också liknande problem som Newtons metod. Speciellt svårt är det att bestämma multipla rötter både med Newtons metod och sekantmetoden (eftersom problemet i detta fall har dålig kondition).

Sekantmetoden kan, liksom Newtons metod, också användas för att beräkna kvadratrötter. Om $f(x) = x^2 - a$ så kan approximationerna till roten beräknas enligt sekantmetoden ur formeln

$$x_{n+1} = x_n - \frac{x_n - x_{n-1}}{x_n^2 - x_{n-1}^2}(x^2 - a) = x_n - \frac{x_n^2 - a}{x_n + x_{n-1}}$$

$$= \frac{x_n^2 + x_n x_{n-1} - x_n^2 + a}{x_n + x_{n-1}} = \frac{x_n x_{n-1} + a}{x_n + x_{n-1}}$$

Den motsvarande iterationsekvationen i Newtons metod är

$$x_{n+1} = \frac{x_n^2 + a}{2x_n} = \frac{x_n x_n + a}{x_n + x_n},$$

så de är rätt så lika.

För $a = 5$ och $x_0 = 2$ får vi de successiva approximationerna $x_1 \approx 2.222222$, $x_2 \approx 2.235941$, $x_3 \approx 2.236070$, $x_4 \approx 2.2360680$, så att konvergensen är i stort sett lika snabb som i Newtons metod.

Man kan också använda kvadratiska approximationer till funktionen $f(x)$. I detta fall kan man tillämpa olika metoder för att öka konvergenshastigheten.

Det finns en variant av sekantmetoden, som kallas regula falsi. I likhet med bisektionsmetoden utgår den från två punkter a och b, för vilka värdena av funktionen f har motsatta förtecken. Funktionen kommer då att ha (åtminstone) en rot inom intervallet.

Om vi antar att a_i och b_i är givna x–värden, så kan vi dra en rät linje som går genom punkterna $(a_i, f(a_i))$ och $(b_i, f(b_i))$. Denna linje är en sekant till kurvan, som har ekvationen

$$y - f(b_i) = \frac{f(b_i) - f(a_i)}{b_i - a_i}(x - b_i).$$

Om c_i är linjens skärningspunkt med x–axeln, så gäller

$$f(b_i) + \frac{f(b_i) - f(a_i)}{b_i - a_i}(c_i - b_i) = 0,$$

varav följer att

$$c_i = \frac{f(b_i)a_i - f(a_i)b_i}{f(b_i) - f(a_i)}.$$

Om $f(a_i)$ och $f(c_i)$ har samma förtecken, väljer vi $a_{i+1} = c_i$ och $b_{i+1} = b_i$, men i motsatt fall sätter vi $a_{i+1} = a_i$ och $b_{i+1} = c_i$ och upprepar proceduren tills roten blivit tillräckligt väl approximerad.

Skillnaden mellan regula falsi och sekantmetoden är, att medan sekantmetoden alltid använder de två senast beräknade punkterna, kommer regula falsi att använda de två punkter, som omger roten. Den skiljer sig sig från bisektionsmetoden, eftersom man där sätter $c_i = (a_i + b_i)/2$.

Regula falsi har, i likhet med bisektionsmetoden, den fördelen att den alltid konvergerar. Men till åtskillnad från sekantmetoden är konvergensen inte superlinjär, utan endast linjär. Metoden är lämplig som startmetod, men inte så bra i närheten av en rot.

I regula falsi kommer den ena ändpunkten i längden förbli konstant under iterationerna, medan den andra blir uppdaterad. Resultatet är att intervallbredden inte går mot noll, såsom i bisektionsmetoden. Om samma ändpunkt förblir densamma, är det dock lätt att korrigera detta t.ex. genom att modofiera funktionsvärdet.