

Linjärisering och Newtons metod

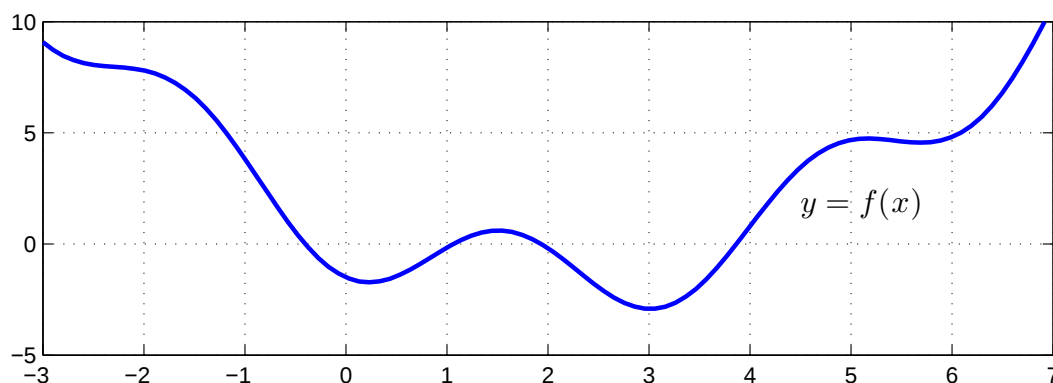
1 Inledning

Vi skall fortsätta med att lösa ekvationer. I förra studioövningen såg vi på intervallhalveringsmetoden. Den är pålitlig men ganska långsam. I denna studioövning skall vi använda Newtons metod som är mycket snabbare, bara vi har en bra första approximation av en lösning. Som exempel kan vi ta,

$$f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5 = 0$$

Vi börjar med att rita grafen till f för att få en uppfattning om hur många nollställen vi har och ungefär var de ligger.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> x=linspace(-3,7);
>> plot(x,f(x))
>> axis([-3 7 -5 10]), grid on
```



Vi ser lösningar till $f(x) = 0$ som de punkter där grafen skär x -axeln. Vi kan grafiskt läsa av en första approximation av en lösning för att sedan förbättra denna med Newtons metod. Innan vi kan komma till Newtons metod måste vi dock först se på linjäriseringar av funktioner i en variabel.

2 Linjärisering

Vi skall se på linjärisering av en differentierbar (deriverbar) funktion i en variabel $f : \mathbb{R} \rightarrow \mathbb{R}$. Linjäriseringen av f runt punkten a ges av (Definition 8, Adams 4.9)

$$L(x) = f(a) + f'(a)(x - a).$$

Nära punkten a har vi $f(x) \approx L(x)$ och den räta linjen $y = L(x)$ är tangenten till kurvan $y = f(x)$ vid a .

Uppgift 1. Linjärisera $f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5$ runt $a = -1$, $a = 2.5$ och $a = 2.8$. Rita funktionskurvan tillsammans med linjäriseringarna i a för de olika a -värdena.

3 Newtons metod

Låt $f : \mathbb{R} \rightarrow \mathbb{R}$ vara en deriverbar funktion. Vi skall lösa ekvationen $f(x) = 0$ med Newtons metod (Adams 4.2) som bygger på successiva linjäriseringar.

Antag att x_k är en approximation av en lösning till ekvationen $f(x) = 0$. Bilda linjäriseringen i x_k , dvs.

$$L(x) = f(x_k) + f'(x_k)(x - x_k)$$

och lös $L(x) = 0$, istället för $f(x) = 0$, och tag denna lösning

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

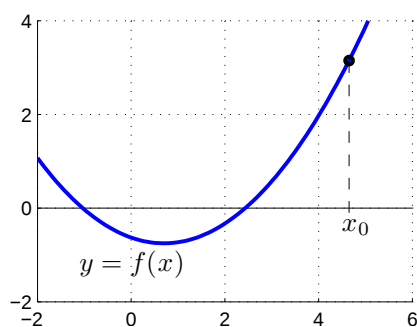
som en ny approximation av nollstället. Detta är Newtons metod.

Som stoppvillkor för iterationen tar vi

$$|x_{k+1} - x_k| \leq \text{tol}$$

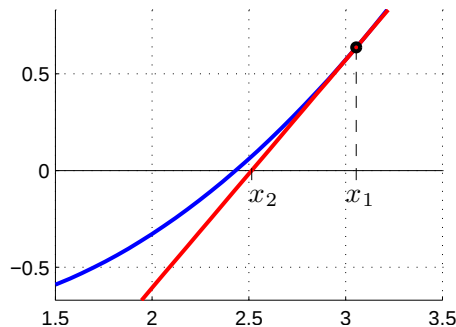
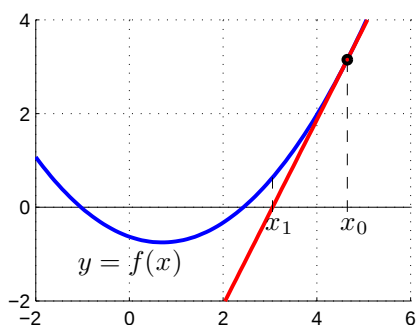
Det betyder att vi accepterar x_{k+1} om ändringen i sista iteration är mindre än toleransen. Vi tillåter maximalt $k_{\max}$ iterationer ($k_{\max} = 10$ är rimligt).

Vi ser på några steg med metoden: Starta med en approximation x_0 av en lösning till $f(x) = 0$.



Bilda tangenten $y = f(x_0) + f'(x_0)(x - x_0)$ till f i $x = x_0$ och tag dess skärningspunkt med x -axeln som en ny approximation

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$



Bilda tangenten $y = f(x_1) + f'(x_1)(x - x_1)$ i $x = x_1$ och tag dess skärningspunkt med x -axeln som en ännu nyare approximation

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

Som exempel tar vi: Lös ekvationen $f(x) = 0$ där

$$f(x) = \frac{1}{2}(x-2)^2 - 2\cos(2x) - \frac{3}{2}$$

Grafen som vi redan ritat visar att vi har ett nollställe nära $x_0 = 4$ som vi tar som startapproximation.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> Df=@(x)x-2+4*sin(2*x);
>> x=4;
>> kmax=10; tol=0.5e-8;
>> for k=1:kmax
    d=-f(x)/Df(x);
    x=x+d;
    disp([x d])
    if abs(d)<tol, break, end
end
```

Utskriften av x_k och $d_k = x_{k+1} - x_k$ ger

```
3.867224680594720   -0.132775319405280
3.866407999346416   -0.000816681248304
3.866407887464428   -0.000000111881988
3.866407887464427   -0.000000000000002
```

För en tillräckligt bra startapproximation x_0 konvergerar Newtons metod mycket snabbt. Ungefär en fördubbling av antalet korrekta decimaler i varje iteration.

Uppgift 2. Låt $f(x) = x^3 - \cos(4x)$. Lös ekvationen $f(x) = 0$. Rita upp grafen till f för att se var ungefär lösningarna (skärningspunkterna) ligger. Hur många lösningar finns det? Läs av i grafiken en första approximation av en lösning för att sedan förbättra denna med Newtons metod. Rita ut lösningen med en liten ring. Upprepa tills du beräknat alla lösningar till ekvationen.

4 Eget program i MATLAB

Det är praktiskt att packetera en metod genom att skriva ett program eller funktion som utför metoden. Vi gjorde det i förra studioövningen för intervallhalveringsmetoden och nu gör vi det även för Newtons metod.

Uppgift 3. Skriv en function som löser ekvationen $f(x) = 0$ med Newtons metod. Funktionen skall heta `min_newton` och skall som indata ges två funktioner, dels en som beräknar $f(x)$ dels en som beräknar $f'(x)$, en startapproximation av lösningen, samt den noggrannhet lösningen skall bestämmas med. Funktionen skall som utdata ge en approximation av nollstället som uppfyller noggrannhetskravet.

Funktionen skall innehålla en hjälptext som beskriver hur den skall användas. Skriver vi `help min_newton` i Command Window så skall det se ut något liknande:

```
>> help min_newton
min_newton - beräknar nollställe till f(x) givet startapproximation x0.
Syntax:
    x = min_newton(f,Df,x0,tol)
Argument:
    f    - funktionshandtag: pekar på namnet till en funktionsfil eller
           till en anonym funktion. T.ex. f=@funk eller f=@(x)cos(x)-x
    Df   - funktionshandtag: pekar på namnet till en funktionsfil eller
           till en anonym funktion som ger derivatan av f.
           T.ex. Df=@Dfunk eller Df=@(x)-sin(x)-1
    x0   - ett tal som ger en startapproximation av nollstället.
    tol  - positivt tal som anger önskad noggrannhet för nollstället.
Returnerar:
    x    - ett tal som ger approximativt nollställe.
Beskrivning:
    Programmet beräknar ett approximativt nollställe till f(x) med
    Newtons metod.
Exempel:
    x = min_newton(@(x)cos(x)-x,@(x)-sin(x)-1,1.0,1e-5)
```

För att underlätta lite finns på studiohemsidan ett programskal `min_newton.m` att utgå ifrån.

Uppgift 4. Använd din funktion `min_newton` för att lösa följande ekvationer. Rita grafer och beräkna samtliga lösningar.

(a). $f(x) = 0.5(x-2)^2 - 2\cos(2x) - 1.5 = 0$ (b). $f(x) = x^3 - \cos(4x) = 0$

5 Modifiering av Newtons metod

En dålig startapproximation kan leda till att Newtons metod divergerar, då är det lämpligt att försöka med dämpad Newton

$$x_{k+1} = x_k - \alpha_k \frac{f(x_k)}{f'(x_k)}, \quad k = 0, 1, \dots$$

Dämpningsfaktorn α_k väljs så att $|f(x_{k+1})| < |f(x_k)|$. Man kan t.ex. börja med $\alpha_k = 1$, på försök ta ett steg i iterationen, om vi har fått en minskning av $|f|$ accepterar vi steget. I annat fall halverar vi successivt α_k och gör nya försök, tills vi har en minskning av $|f|$.

Om vi inte vill beräkna derivator så kan vi approximera dem med differenskvoter

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

för lämpligt valt litet positivt tal h .

6 Färdigt program i MATLAB

Verktyslådan OPTIMIZATION TOOLBOX i MATLAB har en funktion `fzero` som finner nollställan. Metoden i `fzero` är en modifiering av Newtons metod i kombination med intervallhalveringsmetoden, där den senare griper in då startapproximationen inte tillräckligt bra.

Funktion `fzero` används enligt något av alternativen

`x=fzero(fun,x0)` `x=fzero(fun,x0,opts)`

där `fun` beskriver funktionen vi skall finna nollstället till, `x0` är en startapproximation av nollstället eller ett intervall med teckenväxling som omsluter nollstället vi söker. I senare fallet kommer `fzero` garanterat finna en approximation av ett nollställe.

Alternativet med `opts` använder vi då vi t.ex. vill ange hur noggrant lösningen skall beräknas. Man skapar vektorn `opts` med funktionen `optimset`, se hjälptexten.

Vi använder `fzero` för att beräkna en lösning till

$$f(x) = \frac{1}{2}(x-2)^2 - 2\cos(2x) - \frac{3}{2} = 0$$

för en sista gång.

```
>> f=@(x)0.5*(x-2).^2-2*cos(2*x)-1.5;
>> x0=4;
>> x=fzero(f,x0)
x =
    3.866407887464427
```

Uppgift 5. Lös ekvationen $f(x) = (x-5)e^x + 5 = 0$ med `fzero`. Rita graf och se efter var nollställena ligger och beräkna det intressanta nollstället. Läs sedan helt kort om bakgrunden till ekvationen i nästa avsnitt.

7 Svartkroppsstrålning

Detta avsnitt anknyter till Atkins och Jones kapitel 1. Energitätheten vid svartkroppsstrålning ges av Plancks strålningslag

$$u(\lambda, T) = \frac{8\pi hc}{\lambda^5} \frac{1}{e^{hc/k\lambda T} - 1}$$

där λ [m] är våglängden, T [K] är temperaturen, $h = 6.6256 \cdot 10^{-34}$ [Js] är Plancks konstant, $c = 2.9979 \cdot 10^8$ [ms⁻¹] är ljushastigheten i tomrum och $k = 1.3805 \cdot 10^{-23}$ [JK⁻¹] är Boltzmanns konstant.

I nästa studioövning skall vi återskapa figur 1.12, Atkins och Jones kapitel sid 8, som visar graferna av $u(\lambda, T)$ över intervallet $0 < \lambda \leq 10 \mu\text{m}$ för några olika värden på T .

Huvuddelen av strålningen förskjuts mot allt kortare våglängder då temperaturen ökar, se graferna i figur 1.12 (topparna flyttas åt vänster då T ökar). Enligt Wiens förskjutningslag (Atkins och Jones sid 9) gäller sambandet

$$T\lambda_{\max} = b_\lambda$$

där $\lambda_{\max}$ är den våglängd för vilken strålningen är maximal och $b_\lambda = 2.8979 \cdot 10^{-3}$ är Wiens förskjutningskonstant.

Man kan härleda Wiens förskjutningslag från Plancks strålningslag genom att derivera $u(\lambda, T)$ med avseende på λ och sätta derivatan till noll. Vi får

$$\frac{du}{d\lambda} = \frac{d}{d\lambda} \left(\frac{8\pi hc}{\lambda^5} \frac{1}{e^{hc/k\lambda T} - 1} \right) =$$

$$\begin{aligned}
&= -5 \frac{8\pi hc}{\lambda^6} \frac{1}{e^{hc/k\lambda T} - 1} + \frac{8\pi hc}{\lambda^5} \frac{-1}{(e^{hc/k\lambda T} - 1)^2} e^{hc/k\lambda T} \left(-\frac{hc}{k\lambda^2 T} \right) = \\
&= \frac{8\pi hc}{\lambda^6 (e^{hc/k\lambda T} - 1)} \left(\frac{hc}{k\lambda T} \frac{e^{hc/k\lambda T}}{e^{hc/k\lambda T} - 1} - 5 \right)
\end{aligned}$$

så vi har

$$\frac{du}{d\lambda} = 0 \quad \Leftrightarrow \quad \frac{hc}{k\lambda T} \frac{e^{hc/k\lambda T}}{e^{hc/k\lambda T} - 1} - 5 = 0$$

Låter vi $x = \frac{hc}{k\lambda T}$ kommer vi efter förenkling fram till ekvationen

$$f(x) = (x - 5) e^x + 5 = 0$$

Denna ekvation kan inte lösas exakt. I uppgift 5 beräknade vi ekvationens lösning $x^* = 4.9651$ och får därmed $b_\lambda = \frac{hc}{x^* k} = 2.8979 \cdot 10^{-3}$.

Enligt Stefan-Boltzmanns lag (Atkins och Jones sid 9) gäller sambandet $E = \sigma T^4$ mellan totala energimängden E och temperaturen T , där $\sigma = 5.6704 \cdot 10^{-8} \text{ [Wm}^{-2}\text{K}^{-4}\text{]}$ är Stefan-Boltzmanns konstant.

Stefan-Boltzmanns lag kan fås från Plancks lag genom integration, men vi väntar med detta tills läsperiod 2 då vi lär oss om integraler.