Intervallhalveringsmetoden

1 Inledning

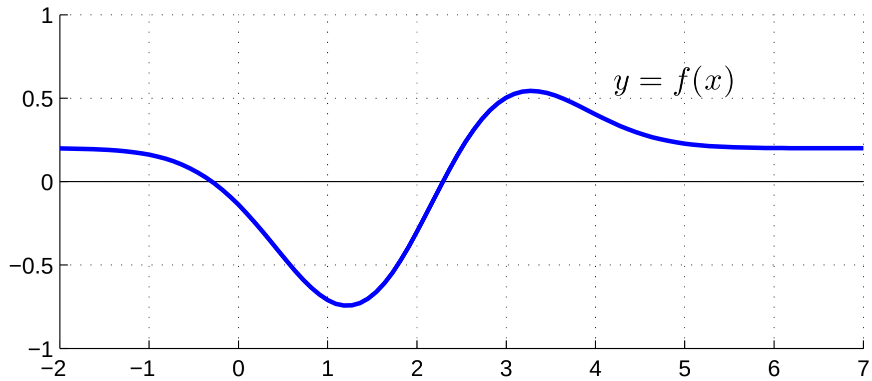
Vi skall lösa ekvationer, dvs. finna nollställen till funktioner. Denna vecka skall vi se på intervallhalveringsmetoden och i nästa vecka skall vi se på Newtons metod. Som exempel kan vi ta ekvationen,

$$f(x) = (x - 2.5)\,\mathrm{e}^{-0.5\,(x-2)^2} + 0.2 = 0$$

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)(x-2.5).*exp(-0.5*(x-2).^2)+0.2;
>> x=linspace(-2,7);
>> plot(x,f(x))
>> axis([-2 7 -1 1]), grid on
```

Vi ser lösningar till $f(x) = 0$ som de punkter där grafen skär x-axeln.



Vi ser att vi har två nollställen, ett i intervallet $[a, b] = [-1, 0]$ och ett i intervallet $[a, b] = [2, 3]$.

Givetvis kan vi grafiskt läsa av en approximation av nollställena, men det är svårt att få hög noggrannhet i denna avläsning.

Vi behöver en metod som stegvis kan förbättra en approximation tills den blir tillräckligt noggrann. Intervallhalveringsmetoden, eller bisektionsmetoden är en sådan metod (The Bisection Method, Adams 1.4).

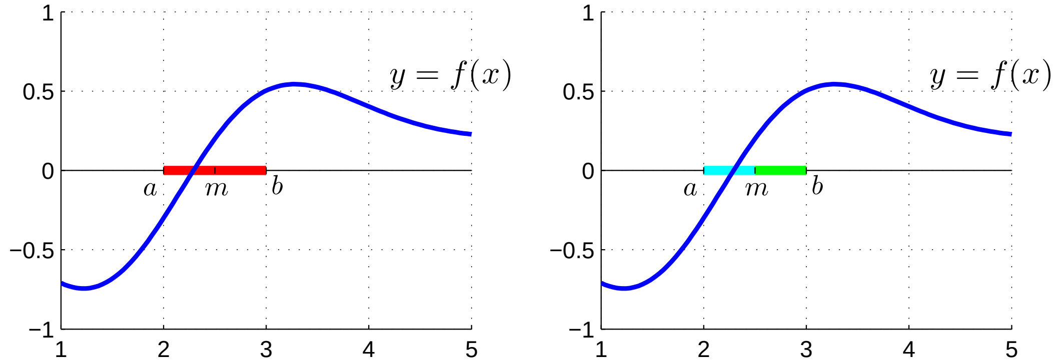
Metoden bygger på följande resonemang: Om en funktion f växlar tecken på ett intervall $[a, b]$ och om f är kontinuerlig så måste f bli noll någonstans i intervallet, dvs. vi har minst ett nollställe i intervallet. Delar vi intervallet i mitten, så att vi får två delintervall, så kommer vi fortfarande ha teckenväxling i något av delintervallen.

Detta delintervall kommer innehålla minst ett nollställe och vi behåller detta intervall och upprepar resonemanget för detta nya intervall.

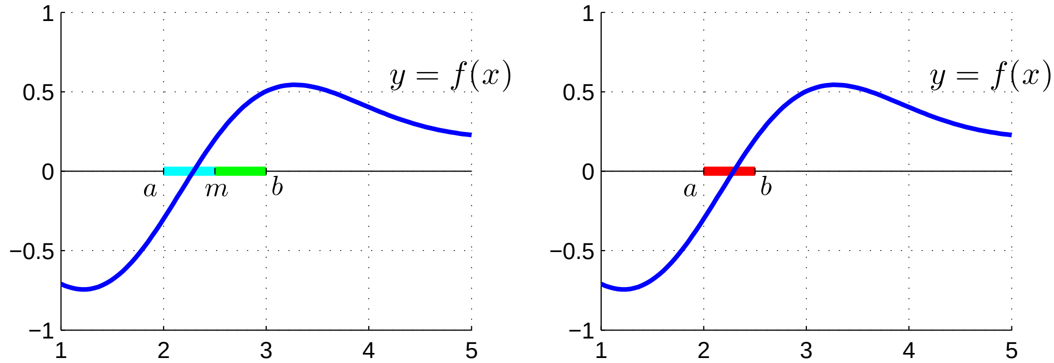
2 Intervallhalveringsmetoden

Antag att funktionen $f(x)$ är kontinuerlig. Starta med ett intervall $[a, b]$ där $f(x)$ växlar tecken, dvs. där $f(a) \cdot f(b) < 0$. Eftersom $f(x)$ är kontinuerlig så finns det minst ett nollställe i $[a, b]$. Bilda mittpunkten $m = \frac{a+b}{2}$ och dela intervallet i två lika långa delintervall $[a, m]$ och $[m, b]$.

Vi delar intervallet i två lika långa delintervall, dvs. $[a, m]$ och $[m, b]$.



Delintervall där $f(x)$ växlar tecken behåller vi, dvs. vi behåller $[a, m]$ om $f(a) \cdot f(m) < 0$, annars behåller vi $[m, b]$.



Upprepa tills vi har ett intervall som är tillräckligt smalt, dvs. att vi har stängt in ett nollställe med tillräcklig noggrannhet.

Uppgift 1. Rita grafen till $f(x) = (x - 2.5)\, \mathrm{e}^{-0.5\,(x-2)^2} + 0.2$, dvs. återskapa bilden i inledningen. Använd en anonym funktion med ett funktionshandtag för att beskriva funktionen (precis som i det inledande exemplet).

Uppgift 2. Vi ser att vi har ett nollställe till $f(x) = 0$ i intervallet $[a, b] = [-1, 0]$.

Skriv följande i MATLAB kod:

(a). Låt $a = -1$ och $b = 0$.

(b). Undersök om f växlar tecken på $[a, b]$ genom att bilda $f(a) \cdot f(b)$.

(c). Bilda mittpunkten m.

(d). Om f växlar tecken på $[a, m]$, låt b få värdet av m, annars låt a få värdet av m. Med andra ord behåll vänster delintervall om vi har teckenväxling där, annars behåll höger delintervall.

Uppgift 3. Skriv en function som löser ekvationen $f(x) = 0$ genom intervallhalvering. Funktionen skall heta `min_bisect` och skall som indata ges en funktion som beräknar $f(x)$, ett intervall $[a, b]$ där

$f(x)$ växlar tecken, dvs. där $f(a) \cdot f(b) < 0$, samt den noggrannhet lösningen skall bestämmas med. Funktionen skall som utdata ge mittpunkten m i det senaste intervallet som omsluter nollstället.

Funktionen skall ge en felutskrift om f inte växlar tecken i intervallet $[a, b]$ som gavs som indata. I detta fall skall m sättas till tomma mängden.

```
??? Felaktig användning ==> min_bisect
 Funktionen måste växla tecken på intervallet.
```

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

```
>> help min_bisect
 min_bisect - beräknar nollställe till f(x) på intervallet I.
   Syntax:
           x = min_bisect(f,I,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
           I   - 1x2 matris, anger ett intervall I=[a,b]. Funktionen måste växla
                 tecken på intervallet.
           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) på
           intervallet I med intervallhalveringsmetoden.
   Exempel:
           x = min_bisect(@(x)cos(x)-x,[0,1],1e-5)
```

För att underlätta lite finns ett programskal `min_bisect.m` på MATLAB-hemsidan att utgå ifrån.

Uppgift 4. Använd nu din funktion `min_bisect` för att beräkna båda nollställena till funktionen i inledningen med fem korrekta decimaler. Kontrollera att svaren är rimliga.

Uppgift 5. Kastbana utan luftmotstånd beskrivs av

$$y(x) = y_0 - \frac{g}{2v_0^2 \cos^2 \theta} \left(x - \frac{v_0^2 \sin(2\theta)}{2g} \right)^2 + \frac{v_0^2 \sin^2 \theta}{2g}$$

där v_0 är utkastfarten och θ är utkastvinkeln.

Hur långt når kastet om vi tar $y_0 = 1.85$, $v_0 = 10$ m/s och $\theta = 45°$? Du känner nog igen funktionen $y(x)$ för kastbanan från tidigare. Gör en funktionsfil för att beskriva den enligt

```
function y=kastbana(x)
y0=1.85; v=10; g=9.81; theta=45; t=theta*pi/180;
a=g/(2*v^2*cos(t)^2); b=v^2*sin(2*t)/(2*g); c=v^2*sin(t)^2/(2*g);
y=y0-a*(x-b).^2+c;
```

Rita en graf av kastbanan så du ser ungefär var marken träffas och använd sedan din funktion `min_bisect` för att bestämma träffpunkten noggrant. Kontrollera att svaret är rimligt.

1 Målsättning

Avsikten med laborationen är dels att få träning i att använda de olika kontrollstrukturerna som vi lärde oss om i förra laborationen, för att skriva ett litet program, dels att se lite på beräkningsmetoder för att lösa ekvationer.

Många ekvationer $f(x) = 0$ i samband med tekniska tillämpningar går inte att lösa med handräkning (det går helt enkelt inte att beskriva lösningen med enkla formler).

Slutligen skall vi också få insikt om behovet av att strukturera arbetet vid problemlösning samt värdet av att dokumentera sitt program. Jobbar man på en utvecklingsavdelning på ett större företag kanske man lägger lika mycket tid på dokumentation som programutveckling.

2 Kommentarer och förklaringar

Intervallhalveringsmetoden utnyttjar kontinuitet och tecknet på funktionsvärden i olika punkter. Hur stort eller lite ett funktionsvärde utnyttjas inte. Metoden fungerar (om vi startar med ett intervall som har teckenväxling) men är inte så effektiv. Den är ändå viktig som en första metod att lösa ekvationer. I nästa laboration skall vi se på Newtons metod som är en mycket effektiv metod, den utnyttjar både funktions- och derivatavärden på ett förståndigt sätt.

3 Lärandemål

Efter denna laboration skall du kunna

- redogöra för den grundläggande idén i intervallhalveringsmetoden
- lösa ekvationer $f(x) = 0$, genom att beskriva f som en function i MATLAB, rita dess graf, grovt lokalisera nollställen av intresse och sedan bestämma dem noggrant med er funktion `min_bisect`