Laboration 1 – Numerisk lösning av ekvationer

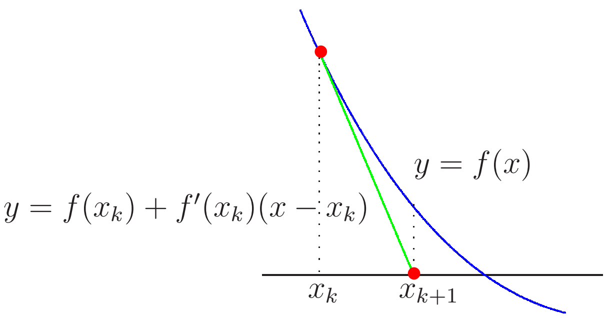
En lösning x^* till en ekvation $f(x) = 0$ kallas ett nollställe eller en rot. Om vi inte kan få fram någon formel för att lösa en ekvation så kan vi beräkna en approximation med t.ex. **Newtons metod**: Antag att x_k är en approximation av ett nollställe till ekvationen $f(x) = 0$. Följ tangenten i punkten $(x_k, f(x_k))$, dvs.

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

ned till x-axeln $(y = 0)$ och tag skärningspunktens x-koordinat

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

som en ny approximation av nollstället.



Som exempel tar vi: Lös ekvationen $f(x) = 0$ där $f(x) = \cos(x) - x$. En graf (rita den gärna) visar att vi har ett nollställe och vi tar $x_0 = 1$ som startapproximation.

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

   0.750363867840244  -0.249636132159756
   0.739112890911362  -0.011250976928882
   0.739085133385284  -0.000027757526078
   0.739085133215161  -0.000000000170123
```

Läs gärna om Newtons metod (eller Newton-Raphsons metod) i Persson-Böiers avsnitt 4.5.

Uppgift 1. Låt $f(x) = x^3 - \cos(4x)$. Lös ekvationen $f(x) = 0$. Rita upp grafen till f för att se var ungefär nollställena ligger. Hur många nollställen finns det? Läs av i grafiken en första approximation av ett nollställe för att sedan förbättra denna med Newtons metod. Rita ut nollstället med en liten ring. Upprepa tills du beräknat alla nollställen till ekvationen.

Uppgift 2. Kastbana utan luftmotstånd (”Mer om funktioner och grafik i MATLAB”, exempel 4) 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 y_0 är utkasthöjden, v_0 är utkastfarten θ är utkastvinkeln och g är tyngdaccelerationen.

Hur långt når kastet om vi tar $y_0 = 1.85$ m, $v_0 = 10$ m/s, $g = 9.81$ m/s^2 och $\theta = 45°$?

Uppgift 3. För att beräkna kvadratroten ur ett tal c, med upprepad addition och division, använde vi iterationsformeln

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{c}{x_k}\right), \; k = 0, 1, 2, \cdots, \; x_0 = c$$

i ”Programmering i MATLAB”, exempel 5. Visa att man får fram formeln genom att använda Newtons metod för att lösa ekvationen $f(x) = x^2 - c = 0$.

Färdiga program i MATLAB

Det finns en färdig funktion `fzero` i MATLAB för att finna nollställen. Som indata tar `fzero` en funktion samt en första approximation av ett nollställe till funktionen och som utdata ger den (förhoppningsvis) en noggrann approximation av detta nollställe.

För polynomekvationer finns funktionen `roots` som tar en vektor med polynomets koefficienter som indata och ger oss alla polynomekvationens rötter som utdata.

Om vi skulle använda `fzero` för att beräkna ett av nollställena i uppgift 1 skulle det kunna se ut så här

```
>> f=@(x)x.^3-cos(4*x);
>> x0=-1;
>> x=fzero(f,x0)
```

Uppgift 4. Rita den kurva som implicit definieras av

$$x^2 + y^2 = 1 + a\sin(xy)$$

för $a = 0.8, 1.6, 2.4$ och 3.2.

Ansätt polära koordinater

$$\begin{cases} x = r\cos(\varphi) \\ y = r\sin(\varphi) \end{cases}, \quad 0 < r, \; 0 \le \varphi < 2\pi$$

Lös ut radien, som funktion av vinkeln, för olika vinklar. Använd `fzero`.

Uppgift 5. Lös polynomekvationen $p(x) = 0$ där

$$p(x) = x^3 - x^2 - x - 1$$

med `roots`. Läs hjälptexten för `roots` först.