

Nemlineáris egyenlet gyökének megkeresése Newton-, szelő- és húrmódszerrel

A feladat továbbra is: $f(x) = 0$ megoldása. Erre a fixpont-módszeren kívül további iterációs eljárások is vannak.

1. Newton-módszer

A Newton-módszernél egy kezdőértékből indulunk ki. Itt meghúzzuk a függvény érintőjét. Az érintő valahol metszi az x tengelyt, ebben a pontban is behúzzuk az érintőt és í.t. Az n -edik lépésben az alábbi képlettel számolunk:

$$x_{n+1} = x_n - f(x_n)/f'(x_n).$$

A metszéspontok sorozata jó esetben konvergál az f zérushelyéhez (pl. ha f konvex, szigorúan monoton növény és a zérushelytől jobbra felvett kezdőértékből indítunk, akkor a metszéspontok monoton csökkenve tartanak a zérushelyhez).

Feladat: oldjuk meg a Matlabban a Newton-módszerrel az $x^3 = x + 2$ egyenletet! A megoldás valahol az $[1, 2]$ intervallumon van. A függvény konvex és szigorúan monoton nő, ezért indítsuk a módszert az $x_0 = 2$ pontból (a zérushelytől jobbra van)!

Az alábbi program elején megadjuk az f függvényt és a deriváltját is. Beállítjuk az előírt pontosságot (pl. ha $f(x_n)$ ennél közelebb van a 0-hoz, akkor leállhat). Beállítunk egy maximális iterációs lépésszámot is arra az esetre, ha túlságosan sok lépést kellene elvégezni a megadott pontosság eléréséhez.

Ezután rögzítjük, hogy melyik pontból indítjuk az iterációt. Egy $\mathbf{x}$ vektorban fogjuk elraktározni a metszéspontokat, ennek a vektornak az első eleme, azaz $\mathbf{x}(1)$ legyen a kezdőpont. Kiértékeljük f -et a kezdőpontban (azért, hogy majd a ciklusban már az első lépésben el tudjuk végezni az összehasonlítást nullával), és bevezetjük az it változót a lépések számolására. A ciklus előtt legyen nulla az értéke, és 1-et akkor vegyen fel, amikor először lép a program a ciklusba. A ciklus egy while ciklus. A feltétel, amelynek teljesülése esetén belép a ciklusba, az, hogy az aktuális pontban még túl nagy a hiba és it nem éri el a maximális iterációs lépésszámot. A cikluson belül it -et 1-gyel megnöveljük, majd kiszámítjuk a következő metszéspontot. Végül kiértékeljük f -et az új metszéspontban. A legvégén a kimenő adatokat definiáljuk: $\mathbf{x_star}$ -ba az egész $\mathbf{x}$ vektort pakoljuk, $\mathbf{f_vertek}$ a legutolsó pontban felvett függvényérték lesz, és $iter$ az it értéke a legvégén.

nl_newton_sima.m:

```
function [x_star f_verték iter] = nl_newton_sima
```

```

% Newton-modszer
% a program akkor all le, ha |f(x_n)| eleg kicsi
f = @(x)(x.^3-x-2); % megadjuk az f fuggvenyt
f_der = @(x)(3*x.^2-1); % es a derivaltjat
hiba = 1e-06; % eloirjuk a pontossagot
maxit = 50; % az iteracios lepesek maximalis szama
x(1) = 2; % ebbol a pontbol inditjuk az iteraciott
y = f(x(1)); % megvizsgaljuk, mennyi itt a fgv. ertek
it = 0; % az it valtozo szamolja majd az iteracios lepeseket
while abs(y) > hiba & it < maxit
    it = it + 1;
    x(it+1) = x(it) - f(x(it))/f_der(x(it));
    y = f(x(it+1));
end
x_star = x;
iter = it;
fvertrek = y;

```

A parancsablakból a következő paranccsal futtathatjuk le a programot:

```

»[x y itszam] = nl_newton_sima

```

Az eredmény:

```

x =
    2.0000    1.6364    1.5304    1.5214    1.5214

y =
    1.7407e-008

itszam =
    4

```

Látjuk, hogy 4 lépés elég volt az előírt pontosságú megoldás előállításához. Vegyük a pontosságot még kisebbre, pl. $1e-10$ -re. Így hány lépés szükséges? Hány lépést kell tennünk, ha 200-ból indítjuk az iterációt?

2. Szelőmódszer

A szelőmódszernél egy intervallumból indulunk ki, amely tartalmazza a gyököt. Az egyik végpontja legyen x_1 , a másik x_2 . Az intervallum két végpontja közé eső húr egyenese valahol elmetshi az x tengelyt, ez lesz az x_3 pont. Most x_3 és x_2 közé rajzolunk be húr, ennek x tengellyel alkotott metszéspontja x_4 , és így tovább, az $i + 1$ -edik ponthoz az $i - 1$ -edik és az i -edik pontra fektetett húr húzzuk be, és ennek az x tengellyel alkotott metszéspontja lesz az $i + 1$ -edik pont:

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

3. Húrmódszer

Hasonlít a szelőmódszerhez, de mindig azt a részintervallumot választjuk, amelynek végpontjaiban ellentétes a függvényérték, így a meghagyott intervallum biztosan tartalmaz gyököt.

Tanulmányozzuk az `nl_newton.m` fájlt, amely mindhárom módszerrel megoldja a feladatot, és még az ún. módosított Newton-módszerrel is, amelyben csak háromlépésenként értékeljük ki a deriváltat. Ebben a programban `x_star` egy tömb, amelynek oszlopai adják a négy különböző módszerrel kiszámított közelítő pontsorozatot, `fverték` és `iter` pedig egy-egy négydimenziós vektor, amely megadja a függvényértékeket és az iterációs lépésszámot sorban mind a négy módszerre a következő sorrendben: Newton-, szelő-, módosított Newton- és húrmódszer. Melyikkel kellett a legkevesebbet lépni?