Közelítő és szimbolikus számítások

9. gyakorlat

Newton és Szelő módszer

Készítette:

Gelle Kitti

Csendes Tibor

Somogyi Viktor

Vinkó Tamás

London András

Deák Gábor

jegyzetei alapján

1. Newton módszer

Tegyük fel, hogy az $f(x) = 0$ egyenlet x^* egyszeres[1], izolált[2] zérushelyét akarjuk meghatározni, és hogy ennek környezetében $f(x)$ differenciálható. Válasszunk ki ebből a környezetből egy x_0 kezdőértéket, majd képezzük az

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

iterációs sorozatot. Ezt az eljárást nevezzük *érintő-* vagy *Newton módszer*nek. Az iterációs képlet geometriai értelmezése az, hogy az aktuális x_k pontban meghatározzuk az $f(x)$ függvény és deriváltja értékét, ezekkel képezzük az adott ponthoz húzott érintőt, és a következő iterációs pontnak az érintőnek a zérushelyét vesszük.

Tudjuk, hogy az adott $P(x_0, y_0)$ ponton átmenő m meredekségű egyenes egyenletét az

$$y - y_0 = m(x - x_0)$$

írja le. Ha éppen a k-adik iterációban járunk, akkor az $(x_k, f(x_k))$ ponthoz tartozó érintőt és annak zérushelyét akarjuk meghatározni. Az érintő egyenes egyenlete a következő:

$$y - f(x_k) = m(x - x_k).$$

Kalkulusból ismerős lehet, hogy az adott pontba húzott érintő meredeksége pont az f függvény deriváltjának értéke az adott pontban, tehát $m = f'(x_k)$. Ez alapján az előbbi képlet:

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

alakban felírható. Mivel az x tengellyel való metszéspontot (a zérushelyet) keressük, ezért $y = 0$. Tehát az előbbi egyenletünkből ezt akár el is hagyhatjuk:

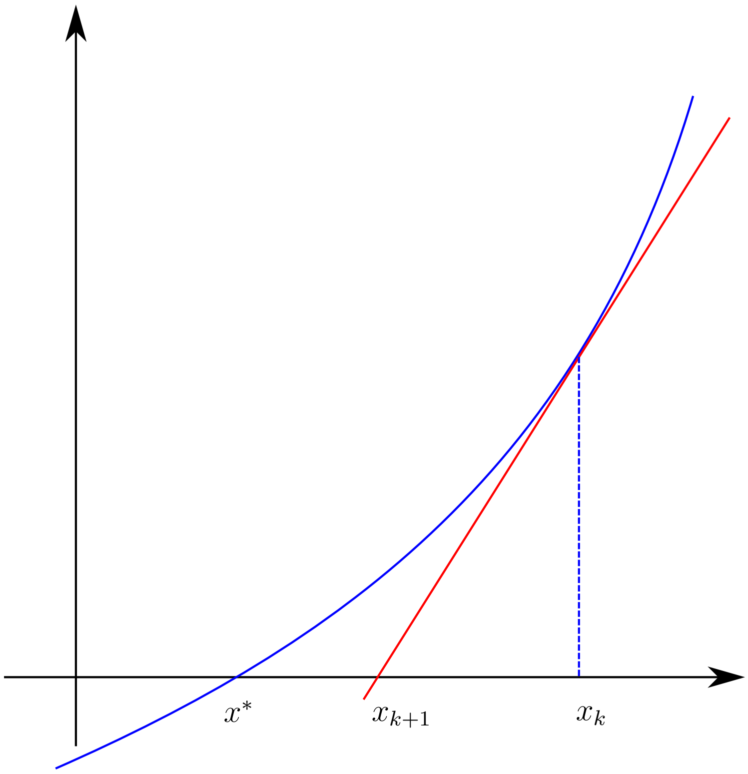
$$-f(x_k) = f'(x_k)(x - x_k).$$

Ezt a képletet átrendezve adódik a Newton módszer képlete:

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

[1] az x^* egyszeres zérushelye f-nek, ha $f'(x^*) \neq 0$

[2] az x^* izolált zérushelye f-nek, ha $f(x^*) = 0$ és van olyan $\epsilon > 0$, hogy az $(x^* - \epsilon, x^* + \epsilon)$ intervallumban x^* az egyetlen zérushely



1. Ábra.: A Newton módszer egy lépése

Példa. Legyen adott az $f(x) = x^3 - x + 1$ függvény. Hajtsunk végre pár iterációs lépést a Newton módszer segítségével.

Megoldás. Határozzuk meg először $f'(x)$-et. Ezt egyszerűen meg tudjuk tenni az ismert derivációs szabályok segítségével: $f'(x) = 3x^2 - 1$. Az iteráció elindításához keresni kell egy környezetet, egy intervallumot, amiben van egy zérushely. Ha megnézzük a függvényt pár helyen, akkor azt kapjuk, hogy $f(-2) = -5$ és $f(-1) = 1$ pontoknál levő függvényértékek ellentétes előjelűek. Mivel a függvény folytonos, így -2 és -1 közt valahol kell legyen egy zérushelye, indítsuk ebből az intervallumból az iterációt, mondjuk legyen $x_0 = -1$.

Írjuk fel a Newton módszer képletét a mostani esetünkben:

$$x_{k+1} = x_k - \frac{x_k^3 - x_k + 1}{3x_k^2 - 1}$$

Behelyettesítve x_0-t, ezt kapjuk x_1-re:

$$x_1 = -1 - \frac{(-1)^3 - (-1) + 1}{3(-1)^2 - 1} = x_1 = -\frac{3}{2} = -1.5.$$

Számoljuk ki x_2-t is:

$$x_2 = -3/2 - \frac{(-3/2)^3 - (-3/2) + 1}{3(-3/2)^2 - 1} = x_1 = -\frac{31}{23} \approx -1.34.$$

A számolást nem folytatjuk tovább. A függvény zérushelye -1.3247, láthatjuk, hogy szépen közelíti ezt az értéket az iteráció.

2. Szelő módszer

A Newton módszer egyik hátránya, hogy szükség van a deriváltak kiszámítására, ami költséges művelet és ha a függvény egyébként nem ismert (csak behelyettesíteni tudunk, de nincs rá explicit képletünk, vagy csak valami közelítő algoritmusunk van), akkor ki se lehet számolni a deriváltat. Általában egyszerűbb, ha $f'(x_k)$ helyett a numerikus deriváltat használjuk:

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

Visszaírva a fentit az iterációs képletbe, a következőt kapjuk:

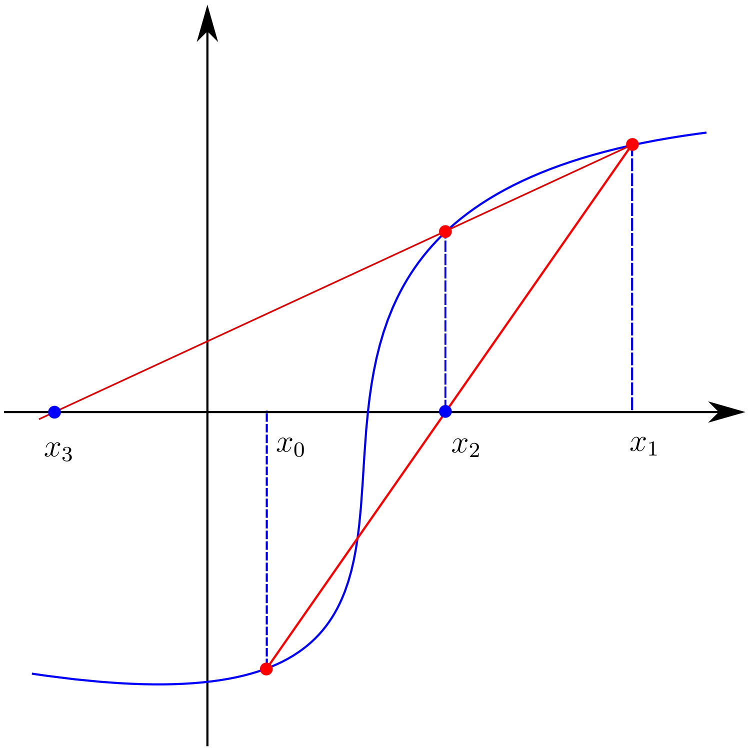
$$x_{k+1} = x_k - \frac{f(x_k)}{\frac{f(x_k)-f(x_{k-1})}{x_k - x_{k-1}}}$$

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

Ennél a módszernél két kezdőértéket kell választani, legyenek ezek x_0 és x_1. Az x_{k+1} nem lesz más, mint az $(x_k, f(x_k))$ és $(x_{k-1}, f(x_{k-1}))$ pontokon átmenő egyenes metszéspontjának x koordinátája az x tengellyel.

A szelő módszert szokás olyan kezdőértékekkel indítani, amelyek a keresett x^* gyököt közrefogják. Másrészt ha $f'(x^*) > 0$, és $f''(x^*) > 0$, akkor x^*-nál nagyobb (de ahhoz közeli) kezdőértékekkel szigorúan monoton konvergencia érhető el.

Azt az algoritmus-változatot, amely felteszi, hogy a kezdeti x_0, x_1 pontokban az $f(x)$ függvény ellentétes előjelű, és $f(x_{k+1})$ előjele függvényében a megelőző két pontból azt választja a következő iterációs lépéshez, amelyikkel ez a tulajdonság fennmarad, *húrmódszer*nek nevezzük.



2. Ábra.: A szelő módszer ábrázolása

Példa. Legyen adott az $f(x) = x^3 - x + 1$ függvény megint. Hajtsunk végre pár iterációs lépést a szelőmódszer segítségével.

Megoldás. Az Newton módszernél kiderítettük, hogy -2 és -1 között megváltozik a függvény előjele, ami azt jelenti hogy ott zérushely van. Most ezt a két értéket használjuk fel úgy, hogy legyen $x_0 = -2$ és $x_1 = -1$. Írjuk fel a szelőmódszer képletét esetünkben:

$$x_{k+1} = x_k - \frac{(x_k^3 - x_k + 1)(x_k - x_{k-1})}{(x_k^3 - x_k + 1) - (x_{k-1}^3 - x_{k-1} + 1)}$$

Behelyettesítve x_0-t és x_1-et, ezt kapjuk x_2-re:

$$x_2 = x_1 - \frac{(x_1^3 - x_1 + 1)(x_1 - x_0)}{(x_1^3 - x_1 + 1) - (x_0^3 - x_0 + 1)}$$

$$x_2 = -1 - \frac{((-1)^3 - (-1) + 1)((-1) - (-2))}{((-1)^3 - (-1) + 1) - ((-2)^3 - (-2) + 1)} = -\frac{7}{6} \approx -1.6667$$

Számoljuk ki x_3-at is. Ehhez már $x_2 = -7/6$-ot és $x_1 = -1$-et fogjuk felhasználni.

$$x_3 = -\frac{7}{6} - \frac{((-7/6)^3 - (-7/6) + 1)((-7/6) - (-1))}{((-7/6)^3 - (-7/6) + 1) - ((-1)^3 - (-1) + 1)} \approx -1.3956$$

A számolást nem folytatjuk tovább.

Megoldás húrmódszerrel. Tudjuk, hogy -2 és -1 pontoknál a függvényérték előjele különböző. Ezért ezt a két értéket használjuk fel kezdőértéknek: $x_0 = -2$ és $x_1 = -1$. Írjuk fel a szelőmódszer képletét esetünkben:

$$x_{k+1} = x_k - \frac{(x_k^3 - x_k + 1)(x_k - x_{k-1})}{(x_k^3 - x_k + 1) - (x_{k-1}^3 - x_{k-1} + 1)}$$

Behelyettesítve x_0-t és x_1-et, ezt kapjuk x_2-re:

$$x_2 = x_1 - \frac{(x_1^3 - x_1 + 1)(x_1 - x_0)}{(x_1^3 - x_1 + 1) - (x_0^3 - x_0 + 1)}$$

$$x_2 = -1 - \frac{((-1)^3 - (-1) + 1)((-1) - (-2))}{((-1)^3 - (-1) + 1) - ((-2)^3 - (-2) + 1)} = -\frac{7}{6} \approx -1.6667$$

Ahhoz, hogy megtudhassuk, melyik pontot kell megtartanunk az előzőek közül x_2 mellé, számoljuk ki a függvényértéket ebben a pontban:

$$f(-\frac{7}{6}) \approx 0.5787$$

. Tehát azt a pontot kell megtartanunk, ahol a függvényérték negatív, ez pedig x_0. Ezek alapján számoljuk ki x_3-at is. Ehhez már $x_2 = -7/6$-ot és $x_0 = -2$-et fogjuk felhasználni.

$$x_3 = -\frac{7}{6} - \frac{((-7/6)^3 - (-7/6) + 1)((-7/6) - (-2))}{((-7/6)^3 - (-7/6) + 1) - ((-2)^3 - (-2) + 1)} \approx -1.2531$$

A számolást nem folytatjuk tovább.

2.1 Konvergencia

A teljesség igénye nélkül a fenti három módszer konvergenciájáról elmondható, hogy általában a Newton módszer gyorsabban konvergál, mint a húrmódszer. A húrmódszer pedig általában gyorsabban konvergál, mint a szelő módszer, ellenben az utóbbi két módszer előnye a Newton módszerhez képest, hogy csak a függvényt kiszámító szubrutinra támaszkodnak és semmi egyéb információra nincs szükségük a függvényről.

3. Polinomok Matlabban

A Matlabban a polinomokat egy vektorban adjuk meg az együtthatóikkal. Például nézzük meg a következő polinomot, és a Matlabos ábrázolását, valamint a gyökeit:

$$p(x) = 4x^3 + 2x^2 - x + 12$$

```
>> p = [4 2 -1 12];
>> r = roots(p)
r =
  -1.6936
  0.5968 + 1.1896i
  0.5968 - 1.1896i
```

Fontos, hogy egy polinom minden hatványának együtthatóját meg kell adjuk, akkor is, ha az nulla! Tehát például a $p(x) = 5x^4 + 3$ polinom a `p = [5 0 0 0 3]` utasítással adható meg.

3.1 Műveletek polinomokkal

A polinomokkal való műveleteket több Matlab utasítás is segíti. A polinomok szorzása és osztása a `conv`[3] és `deconv` paranccsal valósítható meg. Nézzünk rájuk példát:

```
>> p1 = [1 2 -3];
>> p1 = [-1 3 4];
>> conv(p1,p2)
ans =
  [-1 1 13 -1 -12]
```

És tényleg ezt kapjuk:

$$(x^2 + 2x - 3)(-x^2 + 3x + 4) = -x^4 + x^3 + 13x^2 - x - 12$$

.

Az osztás maradékos osztásként is működik, ha az eredményt egy vektorban kérjük:

```
>> [q r] = deconv(p1,p2)
q =
  [-1]
r =
  [0 5 1]
```

[3] A polinomok szorzása az ún. konvolúció művelettel történik

És tényleg ezt kapjuk:

$$\begin{array}{l} (x^2+2x-3) : (-x^2+3x+4) = -1 \\ \underline{x^2-3x-4} \\ \qquad\quad 5x+1 \end{array}$$

A polinomok deriválására is van Matlab utasítás:

```
>> p = [2 -1 3 4];
>> dp = polyder(p)
dp =
  [6 -2 3]
```

Szintén van beépített utasítás polinom egy x^* pontban való kiértékelésére is:

```
>> p = [1 -1 0 0];
>> polyval(p,0.5)
dp =
  -0.1250
```

Ebben az esetben, ha a második argumentum egy vektor vagy mátrix, a kiértékelés minden egyes elemre megtörténik és az eredményt is ugyanebben a formában kapjuk.

3.2 Anonim függvények

Matlab alatt lehetőségünk van úgynevezett anonim függvények használatára. Ezek sok esetben megkönnyítik az életünket, mert egyszerűen és jól látható módon tudunk velük függvényeket reprezentálni a Matlab parancssori felületén. Lássunk is egy példát:

```
>> f = @(x)x.^2; %elemenkenti negyzetre emeles
>> f(2)
ans =

    4
```

A fenti sorban tehát először egy nevet adunk a függvényünknek, majd a '@' jel után megadjuk az argumentumokat, és végül magát a függvény törzsét. Mint látható, ezzel nagyon tömören tudunk *egyszerű* matematikai függvényeket reprezentálni.

További hasznos tulajdonsága, hogy át tudjuk adni különösebb trükk nélkül a saját Matlab függvényeinknek. Tekintsük például a következőt:

```
function [ r ] = FgvTest( f )

r = f(9);

end
```

```
>> g = @(x)x.^2; %elemenkenti negyzetre emeles
>> FgvTest(g)
ans =

    81
```

Itt tehát feltételezzük, hogy a bejövő paraméter egy függvény. Ezzel egy tetszőleges függvényt át tudunk adni paraméterként, ezzel például a Newton vagy Szelő módszer implementálását könnyítve.