Numerikus matematika

Baran Ágnes

Gyakorlat
Nemlineáris egyenletek

Feladatok

(1) Mutassa meg, hogy az $3x^3 - 12x + 4 = 0$ egyenletnek van gyöke a $[0, 1]$ intervallumban. Vizsgálja meg az $x_0 \in [0, 1]$,

$$x_{k+1} = \frac{3x_k^3 + 4}{12}, \quad k = 0, 1, \ldots$$

eljárás konvergenciáját! Írjon egy Matlab-kódot, amely kiszámolja az iteráció első néhány lépését! Módosítsa a kódot úgy, hogy olyan k értékre álljon le, amelyre $|x_k - x_{k-1}| < \varepsilon$, ahol $\varepsilon > 0$ adott.

(2) Mit mondhatunk az $x_0 \in [-\pi/2, \pi/2]$, $x_{k+1} = \frac{1}{3}\cos(x_k)$ eljárás konvergenciájáról?

(3) Közelítse $\sqrt{5}$ értékét Newton-módszerrel, 3 lépésig.

(4) Írjon egy Matlab-függvényt, amely adott függvény zérushelyét közelíti megadott x_0 kezdőpontból indulva, adott ε pontosság esetén. A függvény a zérushely közelítésével, az ottani függvényértékkel és az elvégzett lépések számával térjen vissza.

Feladat

(5) Az előző kód segítségével közelítse az $\ln(x) = 2 - x$ egyenlet gyökét!

(6) Az előző kód segítségével közelítse az $x^3 - 20x = 0$ egyenlet gyökét $x_0 = 2$-ből indulva. Mit tapasztal? Módosítsa az előző Matlab-függvényt úgy, hogy a bemeneti értékek között szerepeljen egy maximális iterációszám is.

(7) Közelítse az $x^3 - 3x - 2 = 0$ egyenlet gyökét Newton–módszerrel az $x_0 = 1.5$ pontból indulva!

(8) Közelítse az $x^3 - 3x + 2 = 0$ egyenlet gyökét Newton–módszerrel az $x_0 = 1.5$ pontból indulva! Ugyanilyen x_0 esetén alkalmazza az

$$x_{k+1} = x_k - 2 \cdot \frac{f(x_k)}{f'(x_k)}$$

iterációt! Mit tapasztal?

Feladat

(9) Közelítse az $f(x) = x - 2\sin(x)$ függvény zérushelyét Newton-iterációval az $x_0 = 1.1$ pontból indulva, 6 lépésig. Mit tapasztal? $\varepsilon = 0.001$ esetén hány lépés után áll le az iteráció? Indítsa el az iterációt $x_0 = 1.4$-ből is. Mit tapasztal? Próbálja megmagyarázni a tapasztalt jelenséget.

Feladat

(10) Tekintsük a

$$\begin{aligned} -4x_1 + \cos(2x_1 - x_2) &= 3 \\ -3x_2 + \sin x_1 &= 2 \qquad x_1, x_2 \in [-\pi, \pi] \end{aligned}$$

egyenletrendszert. Mit mondhatunk az egyenletrendszer megoldhatóságáról, illetve az

$$\begin{aligned} x_1^{(k+1)} &= -\frac{3}{4} + \frac{1}{4}\cos(2x_1^{(k)} - x_2^{(k)}) \\ x_2^{(k+1)} &= -\frac{2}{3} + \frac{1}{3}\sin x_1^{(k)} \end{aligned}$$

$(k = 0, 1, 2, \dots)$ eljárás konvergenciájáról?

Nemlineáris egyenletek Matlab-bal

Az fzero függény.

Az f nemlineáris függvény olyan gyökének közelítésére, melyek környezetében előjelet vált a függvény.

Ha `f` function handle-ként adott, akkor

- `x=fzero(f,x0)`
  Az `x0` pontból indulva előállítja a gyök `x` közelítését
- `[x,fval,exitflag,output]=fzero(f,x0)`
  Az `x0` pontból indulva előállítja a gyök `x` közelítését, megadja az `x` helyen a függvényértéket (`fval`), a közelítőeljárás leállásának az okát (`exitflag`), és a keresési eljárás néhány részletét (`output`)

Példa

Matlabbal számítsuk ki az $f(x) = e^x - 4x^2$ függvény $[0,1]$-beli gyökét.

Megoldás.

```
>> f=@(x) exp(x)-4*x^2;
>> [x,fval]=fzero(f,0.5)
x=
  0.7148
fval=
  -4.4409e-16
```

A gyök közelítése 4 tizedesjegyre kerekítve: 0.7148, ebben a pontban a függvényérték $-4.4409 \cdot 10^{-16}$.

Ha több tizedesjegyre akarjuk látni az eredményt állítsuk át a kiiratás formátumát!

```
>> format long
>> x
x=
 0.714805912362778
```

Ha a keresési eljárásról több információt szeretnénk:

```
>> [x,fval,exitflag,output]=fzero(f,0.5)
x =
   0.714805912362778

fval =
    -4.440892098500626e-16

exitflag =
     1

output =
    intervaliterations: 9
            iterations: 6
             funcCount: 25
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0.273726, 0.726274]'
```

Ha ismerünk olyan intervallumot, ahol a függvény előjelet vált, akkor meggyorsíthatjuk a keresési eljárást, ha `x0` helyett ezzel az intervallummal hívjuk az `fzero` függvényt:

```
>> [x,fval,exitflag,output]=fzero(f,[0,1])
x =
   0.714805912362778

fval =
    -4.440892098500626e-16

exitflag =
     1

output =
    intervaliterations: 0
            iterations: 7
             funcCount: 9
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0, 1]'
```

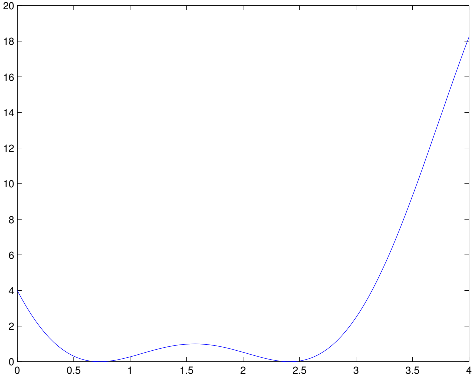
Ha a függvény nem vált előjelet a gyök környezetében, akkor az `fzero` függvény nem találja meg a gyököt:

```
>> f=@(x) 13-9*cos(x).^2-12*sin(x);
>> x=fzero(f,0)
Exiting fzero: aborting search for an interval containing a sign change
    because NaN or Inf function value encountered during search.
(Function value at -Inf is NaN.)
Check function or try again with a different starting value.

x =

   NaN
```

Ábrázoljuk az f függvényt a $[0, 4]$ intervallumon!



Az ábra alapján azt sejtjük, hogy a függvénynek 0.5 és 2.5 környezetében van 1-1 gyöke, ahol a függvény nem vált előjelet. Az is látszik, hogy itt a függvénynek minimuma van.

fminbnd

- `x=fminbnd(f,xmin,xmax)`
- `[x,fval,exitflag,output]=fminbnd(f,xmin,xmax)`

Megkeresi az `f` függvény $[xmin, xmax]$ intervallumbeli minimumát.

Példa:

```
>> f=@(x) 13-9*cos(x).^2-12*sin(x);
>> [x,fval]=fminbnd(f,0,1)
x =
    0.7297
fval =
   9.2491e-11
>> [x,fval]=fminbnd(f,2,3)
x =
    2.4119
fval =
   1.7231e-13
```

Ha az ábra alapján -vagy egyéb rendelkezésre álló információ alapján- úgy látjuk, hogy a függvénynek olyan gyöke van, ahol nem vált előjelet, és ez a függvény lokális maximuma, akkor az `fminbnd` függvényt $-f$-re hívjuk meg.

Fontos! Mindig ellenőrizzük (a függvényérték kiiratásával), hogy az adott szélsőértékhely valóban gyökhely-e.

Polinomok gyökeinek keresésére az `roots` függvényt használhatjuk: keressük meg az $x^4 + x^3 - x - 1$ polinom gyökeit!

```
>> p=[1 1 0 -1 -1];
>> roots(p)
ans =
   1.0000 + 0.0000i
  -0.5000 + 0.8660i
  -0.5000 - 0.8660i
  -1.0000 + 0.0000i
```

Feladatok

(a) Közelítse a $3x = \cos(x)$ egyenlet gyökeit!

(b) Közelítse a $3x^3 - 12x + 4 = 0$ egyenlet gyökeit!

(c) Közelítse az $e^x = \sin(x)$ egyenlet gyökeit!

(d) Közelítse az $\ln(x) = 2 - x$ egyenlet gyökét!

(e) Közelítse a $\cos^2(x) + 2\sin(x) = 2$ egyenlet gyökét!

(f) Közelítse az $x^4 - x^3 - 2x^2 - 2x + 4 = 0$ egyenlet gyökeit!