

## 3. BEADANDÓ PROGRAM

**Csillapított Gauss-Newton módszer.** (A módszer leírását ld. Stoyan Gisbert: Numerikus matematika mérnököknek és programozóknak, 153. oldal.)

Adott  $t_1, \dots, t_m$  alappontok és  $f_1, \dots, f_m$  függvényértékek esetén meg kell határozni egy, az adatokra jól illeszkedő, adott nemlineáris modell paramétereit.

Jelölje  $F(x, t)$  az illesztendő modellt, ahol  $x = (x_1, \dots, x_n)$  a modell ismeretlen paraméterei. Legyen  $G(x) = (F(x, t_1), \dots, F(x, t_m))^T$ , továbbá  $J(x)$  a  $G$  függvény Jacobi mátrixa. Jelölje  $f$  a mért értékek vektorát. Legyen adott az  $\varepsilon$  pontosság és a *maxit* maximális iterációs szám. A modell paramétervektorát egy vektorsorozattal fogjuk közelíteni a következő módon:

legyen adott a paramétervektor egy  $x^{(0)}$  kezdeti közelítése, ha már ismert  $x^{(k)}$ , akkor  $x^{(k+1)}$  értékét a következőképpen határozzuk meg: oldjuk meg a

$$(J_k^T J_k) \cdot \delta x = J_k^T (f - G(x^{(k)}))$$

lineáris egyenletrendszer (azaz határozzuk meg a  $\delta x$  ismeretlen vektort), ahol  $J_k = J(x^{(k)})$ .

Az ismert  $x^{(k)}$  vektor és a most meghatározott  $\delta x$  vektor segítségével legfeljebb 5 próbálkozást teszünk egy alkalmas  $x^{(k+1)}$  vektort meghatározására:

legyen  $y = x^{(k)} + s \cdot \delta x$ , ahol az  $s$  csillapítási paraméter értéke az algoritmus elején 1. Ha

$$\|f - G(y)\|_2 < \|f - G(x^{(k)})\|_2,$$

akkor az  $y$  vektort elfogadjuk következő közelítővektornak:  $x^{(k+1)} = y$ . Ha a fenti egyenlőtlenség nem teljesül, akkor egy kisebb  $s$  értékkel próbálkozunk (de nem engedjük, hogy  $s$  értéke  $10^{-3}$  alá csökkenjen):  $s = \max\{10^{-3}; 0.7 \cdot s\}$  és újra  $y = x^{(k)} + s \cdot \delta x$ . Ha az 5 próbálkozás során nem sikerül alkalmas  $y$  vektort meghatározni, akkor az algoritmust befejezzük, és új  $x^{(0)}$  kezdeti vektorból indítjuk újra. Ha az  $x^{(k)}$  vektorból első próbálkozásra sikerül alkalmas  $y$  vektort találni, akkor  $s$  értékét óvatosan megnöveljük (de nem engedjük 1 fölé):  $s = \min\{1; 1.2 \cdot s\}$ .

Az algoritmus akkor fejeződik be sikeresen, ha valamely  $x^{(k)}$  esetén

$$\|f - G(x^{(k)})\|_2 \leq \varepsilon \cdot (1 + \|f - G(x^{(0)})\|_2)$$

teljesül. Az iterációt legfeljebb *maxit* lépésig folytatjuk.

**Az algoritmus:**

1. Legyen  $s = 1$  és  $g = f - G(x^{(0)})$ . Az  $x^{(0)}$  kezdeti vektor esetén határozzuk meg  $gn := \|g\|_2$  értékét, és legyen  $gn0 = gn$ .
2. k=0:maxit
3. [ megvizsgáljuk a leállási feltételt:

- ha  $gn \leq \varepsilon \cdot (1 + gn_0)$ , akkor az iteráció sikeres volt, kilépünk.
4.  $J := J(x^{(k)})$  és  $w := J^T \cdot g$
  5. Oldjuk meg a  $J^T J \cdot \delta x = w$  egyenletet (Cholesky-felbontással). Szingularitás esetén kilépés.
  6.  $\ell = 1 : 5$
  7.  $[y := x^{(k)} + s \cdot \delta x$  és  $g_{uj} := f - G(y)$ , továbbá  $gn_{uj} := \|g_{uj}\|_2$
  8. Ha  $gn_{uj} < gn$ , akkor elfogadjuk  $y$ -t, esetleg módosítjuk  $s$  értékét, majd kilépünk az  $\ell$  szerinti for-ciklusból:
 
$$x^{(k+1)} := y, g := g_{uj}, gn := gn_{uj}$$
 ha  $\ell = 1$ , akkor  $s = \min\{1; 1.2 \cdot s\}$ ,  
 kilépés az  $\ell$  szerinti ciklusból.
  9. ha a  $gn_{uj} < gn$  feltétel nem teljesül csökkentsük  $s$  értékét:
 
$$s = \max\{10^{-3}; 0.7 \cdot s\}. \quad ]_\ell$$
  10. Ha az 5 lépés során nem sikerült  $x^{(k+1)}$ -et meghatározni, akkor az iteráció sikertelen volt, kilépünk.  $]_k$
  11. Ha elértük a maximális iterációszámot, akkor kilépés.

A programnak 2 előre adott tesztfeladat esetén kell működnie:

**1. tesztfeladat.** (Tankönyv 155. oldal, első tesztfeladat.)

A  $t_i = -2 + 5 \cdot \frac{(i-1)}{11}$ ,  $i = 1, 2, \dots, 12$ , alappontokban az

$$f_i = 0.2 + 0.1 \cdot t_i + 2 \cdot \exp(-2 \cdot (t_i - 1)^2), \quad i = 1, \dots, 12,$$

értékek adottak. Az illesztendő modell:

$$F(x, t) = x_1 + x_2 \cdot t + x_3 \cdot \exp(-x_4 \cdot (t - x_5)^2),$$

tehát a feladatot úgy konstruáljuk, hogy a modell paramétereit „elfelejtjük”, majd a mérési eredményekből próbáljuk azokat meghatározni.

Ekkor adott  $x$  esetén a  $G(x)$  vektor 12 koordinátából áll, az  $i$ -edik koordinátája ( $i = 1, \dots, 12$ ):

$$G_i(x) = F(x, t_i) = x_1 + x_2 \cdot t_i + x_3 \cdot \exp(-x_4 \cdot (t_i - x_5)^2),$$

a Jacobi-mátrix  $12 \times 5$ -ös lesz.

**2. tesztfeladat.**

A

$$t = (0.32; 3.42; 5.15; 7.24; 10.24; 13.26; 15.67; 18.56)$$

alappontokban az

$$f = (8.581; 8.357; 7.903; 7.093; 5.605; 4.018; 2.924; 2.047)$$

értékek adottak. Az illesztendő modell:

$$F(x, t) = x_1 + x_2 \cdot \cos\left(\pi \frac{t - x_3}{20}\right).$$

Itt tehát 8 mérési eredményünk van, 3 ismeretlen paramétert keresünk. A  $G(x)$  vektorfüggvénynek 8 koordinátája van, a Jacobi-mátrix  $8 \times 3$ -as.

**Input:** A beolvasás a standard inputról történik. Az input első értéke megadja, hogy hány feladatot szeretnénk megoldani ( $N$ , ahol  $N \leq 20$ ), a következő 3 érték ( $n, maxit, \varepsilon$ ) közül az első megadja, hogy az első esetben melyik tesztfeladattal szeretnénk dolgozni ( $n = 1$ , vagy  $n = 2$ ), a második a maximális iterációs szám, a harmadik a pontosság. Ezt követi az első feladat esetén a paramétervektor kezdeti értéke ( $x^{(0)}$ ). Utána ugyanezek az adatok következnek a többi feladatra vonatkozóan.

**Output:**  $N$  részből áll: az  $i$ -edik sor az  $i$ -edik feladatra vonatkozó outputot tartalmazza. Ha a kilépés sikeres volt (az algoritmus 3. lépésében a leállási feltétel teljesült), akkor ebbe a sorba a **siker** üzenet után az elvégzett lépések száma, a paramétervektor utolsó értéke és a  $gn$  értéke kerüljön. Ha az algoritmus azért fejeződött be, mert egy  $x^{(k)}$  vektorból 5 próbálkozásra sem sikerült elfogadható  $x^{(k+1)}$  vektort meghatározni (az algoritmus 10. lépése), akkor az adott sorba a **sikertelen** üzenet után az elvégzett lépések száma, a paramétervektor utolsó értéke és a  $gn$  értéke kerüljön. Ha az algoritmus azért fejeződött be, mert a Cholesky-felbontás során szingularitást tapasztaltunk, akkor ebben a sorban a **szingularis** üzenet jelenjen meg. Ha az algoritmus úgy fejeződött be, hogy elértük a maximális iterációs számot, akkor a **maxit** üzenet jelenjen meg. Az outputban a lebegőpontos számok 8 tizedesjegyig legyenek kiírva.

## 1. Példa:

Ha az 1. tesztfeladattal dolgozunk, és  $maxit = 15$ ,  $\varepsilon = 0.000001$ , továbbá  $x^{(0)} = (0.1; 0.2; 1; 1; 1.5)$ , akkor

$$g(x^{(0)}) = (0.29999525; 0.25445641; 0.20819456; 0.16266201; 0.18150800; 0.54537800; 1.20042258; 0.95014688; -0.15542348; -0.62929106; -0.47291831; -0.30472830),$$

ennek normája  $gn0 = 1.90667241$ . A  $w$  vektor:

$$w = (2.24040254; -2.90029804; 0.86598707; -0.30027732; -2.90311536),$$

a lineáris egyenletrendszer megoldása:

$$\delta x = (0.09806482; -0.06125141; 0.25949139; -0.17697287; -0.97265058).$$

Ezután kiszámítjuk az  $y = x^{(0)} + s \cdot \delta x$  vektort (ahol  $s = 1$ ):

$$y = (0.19806482; 0.13874859; 1.25949139; 0.82302713; 0.52734942),$$

az ehhez tartozó  $g_{uj}$  normája:  $gn_{uj} = 1.48256387$ , ami kisebb, mint az előző közelítőponthoz tartozó  $g$  vektor normája, így  $y$ -t elfogadjuk  $x^{(1)}$ -nek. A részeredmények táblázatban:

$k$	$x_1^{(k+1)}$	$x_2^{(k+1)}$	$x_3^{(k+1)}$	$x_4^{(k+1)}$	$x_5^{(k+1)}$	gn
0	0.19806482	0.13874859	1.25949139	0.82302713	0.52734942	1.48256387
1	0.18672399	0.04411758	1.40336607	0.82897062	1.31601808	1.15556566
2	0.34254834	0.20944705	1.39632372	1.66557186	0.73440294	1.10351067
3	0.19741464	0.09575967	1.74866326	1.43851365	1.10814245	0.50606781
4	0.21020567	0.10947672	1.92415406	1.90515715	0.97250779	0.13483111
5	0.20018063	0.09982019	1.99658122	1.99419713	1.00146988	0.00689149
6	0.20000067	0.10000088	1.99998864	1.99998419	0.99999610	0.00001937
7	0.20000000	0.10000000	2.00000000	2.00000000	1.00000000	0.00000000

## 2. Példa:

Ha a 2. tesztfeladattal dolgozunk, és  $maxit = 10$ ,  $\varepsilon = 0.001$ , továbbá  $x^{(0)} = (3; 6; 9)$ , akkor

$$g(x^{(0)}) = (4.34582435; 1.51795113; -0.03284311; -1.67916603; \\ -3.28154310; -3.68806095; -3.07327889; -1.36736015),$$

ennek normája  $gn_0 = 7.72632369$ . A  $w$  vektor:

$$w = (-7.25847676; -7.51840444; -11.20763769),$$

a lineáris egyenletrendszer megoldása:

$$\delta x = (2.20041216; -4.94991579; -3.43089687).$$

Ezután kiszámítjuk az  $y = x^{(0)} + s \cdot \delta x$  vektort (ahol  $s = 1$ ):

$$y = (5.20041216; 1.05008421; 5.56910313),$$

az ehhez tartozó  $g_{uj}$  normája:  $gn_{uj} = 5.49129431$ , ami kisebb, mint az előző közelítőponthoz tartozó  $g$  vektor normája, így  $y$ -t elfogadjuk  $x^{(1)}$ -nek. Az iteráció következő lépésében

$$w = (0.49859942; 6.77193086; -1.46958189), \\ \delta x = (0.00000000; 1.51065881; -13.55775815), \\ y = (5.20041216; 2.56074301; -7.98865502).$$

Az  $y$ -hoz tartozó  $g_{uj}$  normája:  $gn_{uj} = 8.00226813$ , ami nem kisebb, mint az előző közelítőponthoz tartozó  $g$  vektor normája, így  $y$ -t nem fogadjuk el.  $s$  értékét csökkentjük:  $s = 0.7$ , ezzel

$$y = (5.20041216; 2.10754537; -3.92132757),$$

itt már  $gn_{uj} = 4.90208956$ , így ezt elfogadjuk  $x^{(2)}$ -nek. A részeredmények táblázatban:

$k$	$x_1^{(k+1)}$	$x_2^{(k+1)}$	$x_3^{(k+1)}$	gn
0	5.20041216	1.05008421	5.56910313	5.49129431
1	5.20041216	2.10754537	-3.92132757	4.90208956
2	5.20041216	2.33599359	1.09785932	2.18891932
3	5.20041216	3.22922747	0.97701382	0.35245127
4	5.20041216	3.39975683	1.00016780	0.00134546
5	5.20041216	3.39975683	1.00016780	0.00134546

Mivel  $0.00134546 < \varepsilon \cdot (1 + gn_0)$ , így az algoritmus sikeresen fejeződik be.