

# 8. előadás

## Numerikus analízis

Az  $x = G(x)$  ( $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ) és az  $F(x) = 0$  ( $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ ) alakú egyenletrendszerek közelítő megoldási módszerei közül említünk kettőt. Mindkettő a már egydimenzióban megismert valamelyik módszer kiterjesztése többdimenzióra.

Az  $x = G(x)$  egyenletrendszer fixpont iterációját tulajdonképpen már elmondtuk (a lineáris egyenletrendszerekénél is és az egydimenziós eljárásoknál is). Ha a  $D \subseteq \mathbb{R}^n$  zárt tartományra teljesül a

$$G(x) \in D, \quad x \in D$$

és valamilyen indukált normában a

$$\|G(x) - G(y)\| \leq q \|x - y\|, \quad x, y \in D \quad (0 \leq q < 1)$$

kontrakciós feltétel, akkor tetszőleges  $x^{(0)} \in D$  esetén az alábbi algoritmus lineáris sebességgel konvergál az  $x = G(x)$  egyenlet egyetlen megoldásához.

## Algoritmus

### A fixpont iteráció algoritmus

- 1 Input  $x^{(0)}, \varepsilon > 0$ .
- 2  $k = 1, \quad x^{(1)} = G(x^{(0)})$
- 3 **WHILE** kilépési feltétel=hamis **DO**
- 4      $x^{(k+1)} = G(x^{(k)})$
- 5      $k = k + 1$

### *Megjegyzés*

A kilépési feltételek is azonosak az egydimenziós esettel, annyi módosítással, hogy abszolút érték helyett normát veszünk és felső indexet használunk.

Itt is valamely  $x^{(0)} \in \mathbb{R}^n$  kezdeti vektorból kiindulva állítunk elő egy sorozatot, amely reményeink szerint (bizonyos feltételek megléte esetén garantáltan) az egyenletrendszer megoldásához konvergál. Az  $F(x) = 0$  ( $x \in \mathbb{R}^n$ ) egyenletrendszer skalár alakja:

$$f_1(x_1, x_2, \dots, x_n) = 0$$

$$\vdots$$

$$f_n(x_1, x_2, \dots, x_n) = 0.$$

Az egyes skalár egyenleteket linearizáljuk az

$x^{(k)} = [x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}]^T \in \mathbb{R}^n$  pontban, majd az így kapott lineáris egyenletrendszer megoldása szolgáltatja az  $x^{(k+1)}$

közelítést. Az  $i$ -edik skalár függvény elsőfokú Taylor-polinomja az  $x^{(k)}$  pontban:

$$f_i(x_1, x_2, \dots, x_n) \approx f_i(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) + \sum_{j=1}^n \frac{\partial}{\partial x_j} f_i(x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}) (x_j - x_j^{(k)}).$$

Tömörebb formában ugyanez

$$\begin{aligned} f_1(x) &\approx f_1(x^{(k)}) + \nabla f_1(x^{(k)})^T (x - x^{(k)}) \\ &\vdots \\ f_n(x) &\approx f_n(x^{(k)}) + \nabla f_n(x^{(k)})^T (x - x^{(k)}), \end{aligned}$$

ahol  $\nabla f_i(x^{(k)})$  az  $i$ -edik skalár függvény gradiense az  $x^{(k)}$  pontban, azaz az  $F$  Jacobi-mátrixának  $i$ -edik sorvektora az  $x^{(k)}$  pontban. Az  $F(x) = 0$  egyenletrendszer megoldása helyett keressük a linearizált egyenletrendszer

$$\begin{aligned} f_1(x^{(k)}) + \nabla f_1(x^{(k)})^T(x - x^{(k)}) &= 0 \\ &\vdots \\ f_n(x^{(k)}) + \nabla f_n(x^{(k)})^T(x - x^{(k)}) &= 0, \end{aligned}$$

megoldását. Vegyük észre, hogy az  $y = f_i(x^{(k)}) + \nabla f_i(x^{(k)})^T(x - x^{(k)})$  egyenlet az  $y = f_i(x)$  függvény érintő (hiper)síkja az  $x^{(k)}$  pontban. Az  $x^{(k+1)}$  közelítés az érintősíkoknak az  $y = 0$  síkon lévő közös pontja.

Az  $F$  függvény

$$J(x) = \left[ \frac{\partial f_i(x)}{\partial x_j} \right]_{i,j=1}^n = \begin{bmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_n(x)^T \end{bmatrix}$$

Jacobi-mátrixával tömörebben is megfogalmazható a linearizált egyenletrendszer:

$$F(x^{(k)}) + J(x^{(k)})(x - x^{(k)}) = 0. \quad (1)$$

Ennek megoldása:

$$x^{(k+1)} = x^{(k)} - \left[ J(x^{(k)}) \right]^{-1} F(x^{(k)}) \quad (k = 0, 1, \dots). \quad (2)$$

Az  $\{x^{(k)}\}$  sorozat fenti előállítását nevezzük Newton-módszernek. Amennyiben a skalár számmal való osztást úgy tekintjük, hogy az az inverzével (reciprokával) balról történő szorzás, akkor tökéletes analógiát látunk az (2) és az egydimenziós Newton-módszer között. Ugyanígy az algoritmusok között is, annyi eltéréssel, hogy a gyakorlatban soha nem invertáljuk a  $J(x^{(k)})$  Jacobi-mátrixot. Helyette az (1) lineáris egyenletrendszert oldjuk meg valamely alkalmas módszerrel, a  $\Delta^{(k)} = x - x^{(k)}$  új változó bevezetésével.

### *A Newton-módszer algoritmusá egyenletrendszerekre*

- 1 Adott  $x^{(0)} \in \mathbb{R}^n$ ,  $\varepsilon > 0$ .
- 2 **FOR**  $i = 0, 1, 2, \dots$
- 3     Oldjuk meg a  $J(x^{(k)})\Delta^{(k)} = -F(x^{(k)})$  egyenletrendszert.
- 4      $x^{(k)} = x^{(k)} + \Delta^{(k)}$

Az algoritmust itt taxatív ciklussal írtuk le; természetesen gondoskodni kell a kilépési feltételről. Az alkalmazható kilépési feltétel itt is a korábban megadott (B) vagy/és (C) lehet, értelemszerűen valamilyen normában alkalmazva. (Természetesen az iterációszám korlátozásáról se feledkezzünk meg.) Egzakt (A) hibabecslésre igen ritkán van módunk. Ismerünk ugyan konvergenciatételeket, amelyek becslést is adnak a hibára, ám azok feltételeit a legritkább esetben sikerül belátni, illetve az azokban szereplő konstansokat megbecsülni (a költségekről nem is beszélve). Mindenesetre ezen tételek igazolják, hogy az eljárás konvergenciája alkalmas feltételek esetén lokális és másodrendű.

## Megjegyzés

A lokális konvergencián azt értjük, hogy csak bizonyos tartományból, rendszerint az  $x^*$  megoldás szűk környezetéből választott  $x^{(0)}$  kezdeti vektor esetén konvergens. Ebben az értelemben az egy- vagy többdimenziós fixpontiteráció és az egydimenziós Newton-módszer is lokálisan konvergens.

## Példa

Oldjuk meg az  $F(x) = [f_1(x), f_2(x)]^T = 0$ ,  $x = [x_1, x_2]^T$  egyenletrendszert Newton-módszerrel, ha

$$f_1(x) = x_1 + x_2^2 - 5$$

$$f_2(x) = x_1^3 - x_2\sqrt{x_1} - 3.$$

## Megoldás.

A Jacobi-mátrix:

$$J(x) = \begin{bmatrix} 1 & 2x_2 \\ 3x_1^2 - \frac{x_2}{(2\sqrt{x_1})} & -\sqrt{x_1} \end{bmatrix}.$$

## Megoldás

Legyen a kezdővektor  $x^{(0)} = [3, -5]^T$ . A

$$J(x^{(0)})\Delta^{(0)} = -F(x^{(0)})$$

lineáris egyenletrendszer:

$$\begin{bmatrix} 1 & -10.000000 \\ 28.443376 & -1.732051 \end{bmatrix} \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = - \begin{bmatrix} 23.000000 \\ 32.660254 \end{bmatrix}.$$

## Megoldás

Az egyszerűség kedvéért itt elhagytuk a  $\Delta$  vektor felső indexeit.  
A megoldás:

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} -1.014374 \\ 2.198563 \end{bmatrix}, \quad \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} = x^{(0)} + \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} 1.985626 \\ -2.801437 \end{bmatrix}$$

A kilépési feltételhez szükséges normák:

$$\|\Delta\|_{\infty} = 2.198563, \quad \|F(x^{(1)})\|_{\infty} = 8.776312.$$

## Megoldás

A továbbiakban pedig

$$\begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \end{bmatrix} = \begin{bmatrix} 1.384174 \\ -2.046071 \end{bmatrix},$$

$$\|\Delta\|_{\infty} = 0.755366, \quad \|F(x^{(2)})\|_{\infty} = 2.059214.$$

⋮

$$\begin{bmatrix} x_1^{(6)} \\ x_2^{(6)} \end{bmatrix} = \begin{bmatrix} 1.000000 \\ -2.000000 \end{bmatrix},$$

$$\|\Delta\|_{\infty} = 0.16 \times 10^{-4}, \quad \|F(x^{(2)})\|_{\infty} \approx 10^{-9}.$$

Megjegyezzük, hogy a gyök szoros közelében az eljárás ugyanúgy begyorsul, mint egydimenzióban; míg az  $x^{(5)}$  öt tizedesjegyre volt pontos, az  $x^{(6)}$  már valójában 10 jegyre, a 7-lépésben pedig már elértük a lehetséges 15 tizedesjegyű maximális pontosságot és a kilépési normák is  $10^{-9}$ , illetve  $10^{-15}$  nagyságrendűek.