

### 3. Numerikus módszerek

## Iterációs módszerek: Lineáris egyenletrendszerekre és nemlineáris egyenletekre.

**Definíció.** Egy  $m$  egyenletből álló,  $n$ -ismeretlenes *lineáris egyenletrendszer általános alakja*:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ & & & & & & \vdots & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & = & b_m \end{array}$$

Az egyenletrendszer átírható  $A\underline{x} = \underline{b}$  formába is, ahol

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad \underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}, \quad \underline{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Az  $a_{ij}$  együtthatókból képzett  $A$  mátrixot az egyenletrendszer *együtthatómátrixa*. Ha ezt kibővítjük a  $\underline{b}$  vektor  $b_i$  komponenseiből képzett oszlopvektorral, akkor az egyenletrendszer  $m \times (n+1)$ -es *kibővített mátrixát* kapjuk, amit  $A|\underline{b}$ -vel jelölünk.

$$A|\underline{b} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & \dots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_n \end{pmatrix}$$

**Tétel 1.** (Kronecker–Capelli-tétel): Az  $A\underline{x} = \underline{b}$  akkor és csak akkor megoldható, ha az  $A$  együtthatómátrix és az  $A|\underline{b}$  kibővített mátrix rangja megegyezik:  $r(A) = r(A|\underline{b})$ .

**Tétel 2.** Ha az egyenletrendszer megoldható és

- $r(A) < n$ , akkor végtelen sok megoldás van, ha
- $r(A) = n$ , akkor egyértelmű a megoldás.

## Lineáris egyenletrendszerek iterációs módszerei

A lineáris algebrai egyenletrendszerek megoldási módszerei: *direkt* illetve az *iterációs módszerek*. Direkt módszereknek azokat a módszereket nevezzük, amelyekkel pontosan számolva az egyenletrendszer pontos megoldását kapnánk. Ezek közé tartoznak az eliminációs módszerek, a felbontási módszerek és a Cramer-szabály. Hátrányuk, hogy csak kisebb méretű egyenletrendszerek oldhatók meg velük reális időn belül. Így a gyakorlati feladatok esetén (amelyekben gyakran nagy méretű az együtthatómátrix) az iterációs módszereket használjuk, melyek nagy előnye a direkt módszerekkel szemben a kisebb tárigény, valamint kihasználhatók a gyakran meglévő hozzávetőleges információk is a megoldás várható értékeiről.

Tegyük fel, hogy az  $A\underline{x} = \underline{b}$  egyenletrendszer együtthatómátrixa négyzetes ( $A \in \mathbb{R}^{n \times n}$ ) és determinánsa nullától különbözik ( $\det A \neq 0$ ). Ekkor az 2. tételből következően az egyenletrendszernek egyetlen megoldása létezik. Az iterációs módszerek ezen egyértelmű megoldás közelítő meghatározására adnak lehetőséget.

Az iterációs módszerek általában olyan konvergens sorozatot konstruálnak, melyek határértéke az egyenletrendszer megoldása.

A lineáris egyenletrendszert (LER) vektorsorozatokkal közelítjük, törekedve a minél gyorsabb konvergenciára. Az iterációs módszereknek a lényege a következő átalakítás:

$$\boxed{A\underline{x} = \underline{b} \iff \underline{x} = B\underline{x} + \underline{r}}$$

Ilyen alak létezik, sőt nem egyértelmű, hanem sokféle lehet, és a különböző átalakítások szolgáltatják a különféle iterációs módszereket. A  $B$ -t *átmenetmátrixnak* nevezzük.

**Kontrakció:** A  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  függvény *kontrakció*, ha

$$\exists 0 \leq q < 1 : \forall \underline{x}, \underline{y} \in \mathbb{R}^n : \|\varphi(\underline{x}) - \varphi(\underline{y})\| \leq q \|\underline{x} - \underline{y}\|$$

A  $q$  értéket *kontrakciós együtthatónak* nevezzük.

Példa kontrakciónak:  $\frac{1}{2} \cos x, x \in [0, \frac{\pi}{2}] \mid \frac{3-x^2}{3}, x \in [0, 1] \mid 2 + e^{-x}, x \in [1, +\infty)$ .

**Banach-féle fixponttétel:** Legyen  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n$  kontrakció a  $q$  kontrakciós együtthatóval.

Ekkor a következő állítások igazak:

1.  $\exists! \underline{x}^* \in \mathbb{R}^n : \underline{x}^* = \varphi(\underline{x}^*)$ . Azt mondjuk, hogy  $\underline{x}^*$  az  $\varphi$  függvény fixpontja.
2.  $\forall \underline{x}^{(0)} \in \mathbb{R}^n$  kezdőérték esetén az  $\underline{x}^{(k+1)} = \varphi(\underline{x}^{(k)})$  ( $k \in \mathbb{N}$ ) rekurzióval definiált sorozat konvergens, és a határértéke az  $\underline{x}^*$ , azaz

$$\lim_{k \rightarrow \infty} \underline{x}^{(k)} = \underline{x}^*$$

3. teljesül az alábbi hibabecslés:

$$\|\underline{x}^{(k)} - \underline{x}^*\| \leq \frac{q^k}{1-q} \|\underline{x}^{(1)} - \underline{x}^{(0)}\|.$$

Az egyenletrendszer megoldása így az  $\varphi : \mathbb{R}^n \rightarrow \mathbb{R}^n, \varphi(\underline{x}) = B\underline{x} + \underline{r}$  függvény fixpontjának keresésére vezethető vissza.

Ez esetben tehát csak azt kell biztosítani, hogy  $\varphi$  kontrakció legyen, és csak alkalmazni kell rá a fixpont tételt.

$$\|\varphi(\underline{x}) - \varphi(\underline{y})\| = \|B\underline{x} + \underline{r} - (B\underline{y} + \underline{r})\| = \|B\underline{x} - B\underline{y}\| = \|B(\underline{x} - \underline{y})\| \leq \|B\| \|\underline{x} - \underline{y}\|$$

Tehát, ha a  $B$  mátrix valamely indukált vagy illeszkedő normája kisebb egynél, akkor az  $\varphi$  függvény kontrakció, és a fixponttétel alkalmazható.

**Tétel: (*Elégséges feltétel a konvergenciára*):** Ha a lineáris egyenletrendszer  $B$  átmenetmátrixára  $\|B\| < 1$ , akkor tetszőleges  $\underline{x}^{(0)}$ -ból indított  $\underline{x}^{(k+1)} := B\underline{x}^{(k)} + \underline{r}$  iteráció konvergál az  $A\underline{x} = \underline{b}$  lineáris egyenletrendszer megoldásához.

A konvergencia vizsgálatokor lehet, hogy a fenti elégséges feltétel nem teljesül, ezért nézzük a konvergencia szükséges és elégséges feltételét.

💡 ▶ **Lemma.** Legyen  $B \in \mathbb{R}^{n \times n}$ . Ekkor  $\forall \varepsilon > 0$  esetén  $\exists \|B\|$  indukált norma, úgy hogy

$$\|B\| < \rho(B) + \varepsilon$$

ahol  $\rho(B)$  a  $B$  mátrix spektrálsugara, vagyis

$$\rho(B) = \inf_{\|\cdot\| \text{ indukált}} \|B\| \quad \forall B \in \mathbb{R}^{n \times n}$$

◁ 💡

**Tétel: (*Szükséges és elégséges feltétel a konvergenciára*):** Tetszőleges  $\underline{x}^{(0)} \in \mathbb{R}^n$ -ből indított  $\underline{x}^{(k+1)} := B\underline{x}^{(k)} + \underline{r}$  iteráció konvergál az  $A\underline{x} = \underline{b}$  lineáris egyenletrendszer megoldásához  $\iff \rho(B) < 1$ .

Tehát a következőkben a különböző iterációs eljárások esetén az  $\underline{x} = B\underline{x} + \underline{r}$  átalakítás után képezzük az  $\underline{x}^{(k+1)} = B\underline{x}^{(k)} + \underline{r}$  iterációt valamilyen  $\underline{x}^{(0)}$  kezdőértékkel.

## Jacobi-iteráció

Vegyük az  $A\underline{x} = \underline{b}$  egyenletrendszert, és tegyük fel, hogy  $A \in \mathbb{R}^{n \times n}$  és  $\det(A) \neq 0$ . Ekkor az egyenletrendszernek létezik egyértelmű megoldása.

Írjuk fel az egyenletrendszer komponensenkénti alakját:

$$A\underline{x} = \underline{b} \Leftrightarrow a_{i,1}x_1 + \dots + a_{i,n}x_n = b_i \quad (i = 1, \dots, n) \quad (1)$$

Tegyük fel, hogy  $a_{ii} \neq 0 \quad (i = 1, \dots, n)$ .

Ekkor az (1) átírható  $a_{ii}$ -vel leosztva és átrendezve:

$$x_i = - \left[ \frac{a_{i,1}}{a_{i,i}}x_1 + \frac{a_{i,2}}{a_{i,i}}x_2 + \dots + \frac{a_{i,i-1}}{a_{i,i}}x_{i-1} + \frac{a_{i,i+1}}{a_{i,i}}x_{i+1} + \dots + \frac{a_{i,n}}{a_{i,i}}x_n \right] + \frac{b_i}{a_{i,i}} \quad (\forall i = 1, \dots, n)$$

Ebből felépíthető egy iteráció, ahol  $x_i^{(k)}$  az  $i$ -edik ismeretlen  $k$ -adik közelítése:

$$x_i^{(k+1)} = - \left[ \frac{a_{i,1}}{a_{i,i}}x_1^{(k)} + \dots + \frac{a_{i,n}}{a_{i,i}}x_n^{(k)} \right] + \frac{b_i}{a_{i,i}} \quad \text{ahol } x_i^{(0)} \text{ tetszőleges érték és } k = 0, 1, \dots$$

Összevont alakban:

$$\underline{x}^{(k+1)} := - \frac{1}{a_{i,i}} \left( \sum_{j=1, j \neq i}^n a_{i,j} \underline{x}_j^{(k)} - \underline{b}_i \right) \quad (i = 1, \dots, n)$$

Az iteráció felírható mátrixos alakban is. Továbbra is  $\det(A) \neq 0$ . Ekkor létezik legalább egy elemi szorzata a mátrixnak - a determináns definíciója - szerint, amelyik nem nulla, így pusztán sorcserével elérhető, hogy az  $A$  mátrix diagonálisában csupa nemnulla elem álljon.

Ekkor az  $A$  mátrixnak elkészíthető a következő felbontása

$$A = L + D + U$$

ahol  $L$ : alsó háromszög mátrix,  $D$ : diagonális mátrix,  $U$ : felső háromszög mátrix.

Ekkor azonos átalakításokkal a következő relációk érvényesek:


$$\begin{aligned} A\underline{x} &= \underline{b} \Leftrightarrow (L + D + U)\underline{x} = \underline{b} \\ D\underline{x} &= -(L + U)\underline{x} + \underline{b} \\ \underline{x} &= -D^{-1}(L + U)\underline{x} + D^{-1}\underline{b} \end{aligned}$$

Ebből már származtatható a Jacobi-iteráció mátrixos alakja:

$$\underline{x}^{(k+1)} = \underbrace{-D^{-1}(L + U)}_{:=B_J} \underline{x}^{(k)} + \underbrace{D^{-1}\underline{b}}_{:=r_J}$$

$B_J$  a Jacobi-iteráció *iterációs mátrixát* jelöli.

## A Jacobi-iteráció konvergenciája

 **Definíció.** Az  $A$  mátrix *szigorúan diagonálisan domináns*, ha minden sorban a főátlóban lévő eleme nagyobb, mint az adott sorban lévő összes többi elem abszolút értékének összege, azaz

$$|a_{i,i}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{i,j}| \quad (i = 1, 2, \dots, n)$$

**Tétel:** Ha az  $Ax = b$  lineáris egyenletrendszer mátrixa szigorúan diagonálisan domináns akkor a Jakobi-iteráció konvergens.

A Jakobi iteráció konvergál, ha  $\|B_J\| < 1$ , valamely indukált vagy illeszkedő normában (elégséges feltétel).

## Gauss-Seidel iteráció

A Gauss–Seidel-iteráció abban különbözik a Jacobi-iterációtól, hogy a  $(k+1)$ -edik közelítés  $i$ -edik komponensének kiszámításához felhasználjuk a  $(k+1)$ -edik közelítés már addig kiszámolt komponenseit, vagyis az  $x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}$  értékeket:

$$x_i = -\left[\frac{a_{i1}}{a_{ii}}x_1^{(k+1)} + \dots + \frac{a_{i,i-1}}{a_{ii}}x_{i-1}^{(k+1)}\right] + \left[\frac{a_{i,i+1}}{a_{ii}}x_{i+1}^{(k)} + \dots + \frac{a_{in}}{a_{ii}}x_n^{(k)}\right] + \frac{b_i}{a_{ii}} \quad (\forall i = 1, \dots, n)$$

Összevont alakban:

$$\underline{x}_i^{(k+1)} = -\frac{1}{a_{ii}} \left( \sum_{j=1}^{i-1} a_{i,j} \underline{x}_j^{(k+1)} + \sum_{j=i+1}^n a_{i,j} \underline{x}_j^{(k)} - b_i \right) \quad (i = 1, \dots, n)$$

A módszer a Jacobi-iterációhoz hasonló elven felírható mátrixos alakban.

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \Leftrightarrow (L + D + U)\mathbf{x} = \mathbf{b} \\ (L + D)\mathbf{x} &= -U\mathbf{x} + \mathbf{b} \\ \mathbf{x} &= -(L + D)^{-1}U\mathbf{x} + (L + D)^{-1}\mathbf{b} \end{aligned}$$

Ebből már származtatható a Gauss-Seidel mátrixos alakja:

$$\boxed{\mathbf{x}^{(k+1)} := \underbrace{-(L + D)^{-1}U}_{:=B_{G-S}}\mathbf{x}^{(k)} + \underbrace{(L + D)^{-1}\mathbf{b}}_{r_{G-S}}}$$

$B_{G-S}$  a Jacobi-iteráció iterációs mátrixát jelöli.

**Megjegyzés:** A G-S implementáció során elég egyetlen  $\mathbf{x}$  vektort eltárolni, és annak a komponenseit sorban felülírni, ugyanis láthatjuk, hogy az első  $i - 1$  komponenst már az "új"  $\mathbf{x}^{(k+1)}$  vektorból vesszük.

**Tétel:** Ha  $A \in R^{n \times n}$  és  $A\mathbf{x} = \mathbf{b}$  szigorúan diagonálisan domináns

- $\|B_{G-S}\|_{\infty} \leq \|B_J\|_{\infty} < 1$ . (soraira)
- $\|B_{G-S}\|_1 \leq \|B_J\|_1 < 1$ . (oszlopaira)

Azaz a Gauss-Seidel is konvergens, és legalább olyan gyors, mint a Jacobi.

## Relaxációs módszerek

Vannak esetek, amikor az iterációs mátrix spektrálsugara nem kisebb 1-nél, és ilyenkor nem vagy csak nagyon lassan konvergál az iteráció. Ekkor azt tehetjük, hogy bevezetünk egy paramétert az iterációba, melyet úgy választunk meg, hogy az iteráció konvergens legyen. Ilyenkor egy plusz  $\omega$ , úgynevezett relaxációs paraméter bevezetésével próbáljuk finomítani a módszert.

### A relaxált Jacobi-iteráció

Tekintsük a  $D\mathbf{x} = -(L + U)\mathbf{x} + \mathbf{b}$  egyenletet, valamint a triviális  $D\mathbf{x} = D\mathbf{x}$  egyenletet. Ezeket rendre szorozzuk meg  $\omega$ , illetve  $1 - \omega$  értékekkel, majd adjuk össze a két egyenletet:

$$\begin{aligned} D\mathbf{x} &= (L + U)\mathbf{x} + \mathbf{b} \quad \xRightarrow{\cdot\omega} \quad \omega D\mathbf{x} = -\omega(L + U)\mathbf{x} + \omega\mathbf{b} \quad (1) \\ D\mathbf{x} &= D\mathbf{x} \quad \xRightarrow{\cdot(1-\omega)} \quad (1 - \omega)D\mathbf{x} = (1 - \omega)D\mathbf{x} \quad (2) \\ &\quad (1) + (2) \Downarrow \\ (1 - \omega)D\mathbf{x} - \omega D\mathbf{x} &= (1 - \omega)D\mathbf{x} - \omega(L + U)\mathbf{x} + \omega\mathbf{b} \\ &\quad \Updownarrow \\ D\mathbf{x} - \omega D\mathbf{x} + \omega D\mathbf{x} &= (1 - \omega)D\mathbf{x} - \omega(L + U)\mathbf{x} + \omega\mathbf{b} \\ &\quad \Updownarrow \\ D\mathbf{x} &= (1 - \omega)D\mathbf{x} - \omega(L + U)\mathbf{x} + \omega\mathbf{b} \end{aligned}$$

Szorozzunk  $D^{-1}$ -zel, majd felépítve az iterációt, megkapjuk a módszer mátrixos alakját:

$$\mathbf{x} = (1 - \omega)I\mathbf{x} - \omega D^{-1}(L + U)\mathbf{x} + \omega D^{-1}\mathbf{b} \Leftrightarrow \boxed{\mathbf{x} = \underbrace{((1 - \omega)I - \omega D^{-1}(L + U))}_{B_J(\omega)}\mathbf{x} + \underbrace{\omega D^{-1}\mathbf{b}}_{r_J(\omega)}}$$

$\omega = 1$  esetén a Jacobi-iterációt kapjuk vissza.

Koordinátás alakban felírva:

$$\begin{aligned} x &= (1 - \omega)\underline{x} - \omega(D^{-1}(L + U)\underline{x} + D^{-1}\underline{b}) \\ x &= (1 - \omega)\underline{x} - \omega D^{-1}((L + U)\underline{x} + \underline{b}) \end{aligned}$$

$$\boxed{\underline{x}_i^{(k+1)} = (1 - \omega)\underline{x}_i^{(k)} - \frac{\omega}{a_{i,i}} \left[ \sum_{\substack{j=1 \\ j \neq i}}^n a_{i,j} \underline{x}_j^{(k)} - b_i \right] \quad (i = 1, \dots, n)}$$

**Tétel.** Ha az  $A$  mátrix szimmetrikus és szigorúan diagonálisan domináns, akkor a relaxált Jacobi-módszer  $\omega \in (0, 1]$  esetén konvergens.

### A relaxált Gauss–Seidel-iteráció

Tekintsük a  $(L + D)\underline{x} = -U\underline{x} + \underline{b}$  egyenletet, valamint a triviális  $D\underline{x} = D\underline{x}$  egyenletet. Ezeket rendre szorozzuk meg  $\omega$ , illetve  $1 - \omega$  értékekkel, majd adjuk össze a két egyenletet:

$$\begin{aligned} (L + D)\underline{x} &= -U\underline{x} + \underline{b} \quad \xRightarrow{\cdot \omega} \quad \omega(L + D)\underline{x} = -\omega U\underline{x} + \omega \underline{b} \quad (1) \\ D\underline{x} &= D\underline{x} \quad \xRightarrow{\cdot (1-\omega)} \quad (1 - \omega)D\underline{x} = (1 - \omega)D\underline{x} \quad (2) \\ (1) + (2) &\Downarrow \\ (1 - \omega)D\underline{x} + \omega(L + D)\underline{x} &= (1 - \omega)D\underline{x} - \omega U\underline{x} + \omega \underline{b} \\ &\Updownarrow \\ D\underline{x} - \omega D\underline{x} + \omega L\underline{x} + \omega D\underline{x} &= (1 - \omega)D\underline{x} - \omega U\underline{x} + \omega \underline{b} \\ &\Updownarrow \\ (D - \omega L)\underline{x} &= (1 - \omega)D\underline{x} - \omega U\underline{x} + \omega \underline{b} \end{aligned}$$

Innen  $(D + \omega L)^{-1}$ -gyel átszorozva, majd felépítve az iterációt megkapjuk a módszer mátrixos alakját.

$$\underline{x} = (D + \omega L)^{-1} \left[ (1 - \omega)D - \omega U \right] \underline{x} + \omega(D + \omega L)^{-1} \underline{b}$$

$$\boxed{\underline{x}^{(k+1)} = \underbrace{(D + \omega L)^{-1} \left[ (1 - \omega)D - \omega U \right]}_{B_{G-S}(\omega)} \underline{x} + \underbrace{\omega(D + \omega L)^{-1} \underline{b}}_{r_{G-S}(\omega)}}$$

A koordinátás alak felírásához itt is átírjuk kicsit az iterációt:

$$\begin{aligned} (D + \omega L)\underline{x}^{(k+1)} &= (1 - \omega)D\underline{x}^{(k)} - \omega U\underline{x}^{(k)} + \omega \underline{b} \\ D\underline{x}^{(k+1)} &= -\omega L\underline{x}^{(k+1)} - \omega U\underline{x}^{(k)} + \omega \underline{b} + (1 - \omega)D\underline{x}^{(k)} \\ \underline{x}^{(k+1)} &= -\omega D^{-1} [L\underline{x}^{(k+1)} + U\underline{x}^{(k)} - \underline{b}] + (1 - \omega)\underline{x}^{(k)} \end{aligned}$$

$$\boxed{\underline{x}_i^{(k+1)} = -\frac{\omega}{a_{i,i}} \left[ \sum_{j=1}^{i-1} a_{i,j} \underline{x}_j^{(k+1)} + \sum_{j=i+1}^n a_{i,j} \underline{x}_j^{(k)} - b_i \right] + (1 - \omega)\underline{x}_i^{(k)} \quad (i = 1, \dots, n)}$$

$\omega = 1$  esetén a Gauss-Seidel iterációt kapjuk.



**Definíció:** Az  $A^{n \times n} = [a_{i,j}]$  négyzetes mátrix **szimmetrikus**, ha

$$A^T = A \text{ (a mátrix egyenlő a transzponáltjával, azaz) } a_{i,j} = a_{j,i} \quad \forall i, j = 1, \dots, n \text{ indexre}$$

**Definíció:** Az  $A^{n \times n}$  mátrix pozitív definit, ha  $\forall x \in \mathbb{R}^n \neq 0$  vektorra  $x^T A x > 0$ .

**Definíció:** Az  $A^{n \times n}$  mátrix tridiagonális, ha csak a főátlón és a mellette található két átló mentén vannak nullától különböző elemek.



**Tétel:** Ha  $A$  szimmetrikus és pozitív definit és  $\omega \in (0, 2)$ , akkor a relaxációs módszer konvergens. Ennek következménye a Gauss-Seidel iteráció konvergenciája ( $\omega = 1$  eset).

**Megjegyzés:** Ha  $\omega \notin (0, 2)$ , akkor általában nem konvergens a módszer (bár adott feladat esetén előfordulhat, hogy találunk olyan kezdővektort, amelyből indítva konvergál a módszer).

**Tétel:** Ha  $A$  tridiagonális, akkor  $\rho(B_S) = \rho(B_J)^2$ , azaz a Jacobi és Gauss-Seidel iteráció egyszerre konvergens, illetve divergens.

**Tétel:** Ha  $A$  szimmetrikus, pozitív definit és tridiagonális, akkor a  $J(1)$ ,  $GS(1)$  és  $GS(\omega)$   $\omega \in (0, 2)$ -re konvergens, és  $GS(\omega)$ -ra az optimális paraméter értéke:

$$\omega_0 = \frac{2}{1 + \sqrt{1 - \rho(B_J)^2}}$$

## Richardson-iteráció

Tekintsük az  $A\underline{x} = \underline{b}$  lineáris algebrai egyenletrendszert, ahol  $A$  szimmetrikus, pozitív definit mátrix (azaz minden sajátértéke valós, sőt pozitív). Szorozzuk meg az egyenletrendszer mindkét oldalát egy tetszőleges  $\omega \neq 0$  számmal, majd rendezzük át:

$$\begin{aligned} A\underline{x} = \underline{b} &\iff 0 = -A\underline{x} + \underline{b} \\ &\iff (\cdot \omega) \\ 0 \cdot \omega &= -\omega A\underline{x} + \omega \underline{b} \\ &\iff \\ \underline{x} &= (I - \omega A)\underline{x} + \omega \underline{b} \end{aligned}$$

Az iteráció tehát

$$\boxed{\underline{x}^{(k+1)} := \underbrace{(I - \omega A)\underline{x}^{(k)}}_{B_{R(\omega)}} + \underbrace{\omega \underline{b}}_{r_{R(\omega)}}$$

**Tétel:** Ha  $A \in \mathbb{R}^{n \times n}$  szimmetrikus, pozitív definit, a sajátértékei pedig a következők:

$$0 < m := \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n =: M$$

akkor  $\omega \in (0, \frac{2}{M})$  esetén

$$\rho(I - \omega A) < 1$$

ezért a  $R(\omega)$  (Richardson-iteráció) konvergens.

A  $\rho(I - \omega A)$  spektrálsugár akkor a legkisebb (ezért a konvergencia akkor a leggyorsabb), ha

$$\omega_{opt} = \frac{2}{m + M}$$

Továbbá igaz, hogy:  $\rho(B_{R(\omega_{opt})}) = \frac{M - m}{M + m}$ .

*Megjegyzés:* Ha az  $\frac{M}{m}$  hányados (az  $A$  mátrix kondíciószáma) nagy, akkor a konvergencia még optimális paraméterválasztás mellett is lassú.

## Nemlineáris egyenletek iterációs módszerei

Eddig egyenletrendszerekkel foglalkoztunk, melyekben minden egyenlet lineáris volt. Most módszereket fogunk keresni az  $f(x) = 0$  típusú egyenletek megoldására, ahol  $f \in \mathbb{R} \rightarrow \mathbb{R}$ . A módszerek lényege az lesz, hogy valamilyen szempont szerint egy számsorozatot állítunk elő, melyek bizonyos feltételek mellett az egyenlet gyökéhez konvergálnak.



**Bolzano-tétel:** Legyen  $f \in C[a, b]$  és  $f(a)f(b) < 0$ , azaz az  $f$  függvény az  $a$  és  $b$  pontokban nem 0, valamint ellenkező előjelű. Ekkor létezik  $(a, b)$  intervallumbeli gyöke az  $f$ -nek, azaz  $\exists x^* \in (a, b) : f(x^*) = 0$ .

**A Bolzano-tétel következménye:** Ha a Bolzano-tétel feltételei mellett még  $f$  szigorúan monoton is, akkor az  $x^*$  egyértelműen létezik (hiszen  $f$  invertálható).

**Brouwer-féle fixponttétel:** Legyen  $f : [a, b] \rightarrow [a, b]$  és  $f \in C[a, b]$ .

Ekkor  $\exists x^* \in [a, b] : x^* = f(x^*)$ .

**Tétel:** Legyen  $f : [a, b] \rightarrow [a, b]$ ,  $f \in C^1[a, b]$  és  $f'$  állandó előjelű.

Ekkor  $\exists! x^* \in [a, b] : x^* = f(x^*)$ .

**Fixponttétel [a,b]-re:** Legyen  $f : [a, b] \rightarrow [a, b]$  kontrakció a  $q$  kontrakciós együtthatóval. Ekkor:

1.  $\exists! x^* \in [a, b] : x^* = f(x^*)$ ,
2.  $\forall x_0 \in [a, b] : x_{k+1} = f(x_k)$  konvergens és  $x^* = \lim_{k \rightarrow \infty} x_k$ ,
3.  $|x_k - x^*| \leq q^k |x_0 - x^*| \leq q^k (b - a)$ .

**p-adrendű konvergencia:** Az  $(x_k)$  konvergens sorozat ( $\lim_{k \rightarrow \infty} x_k = x^*$ )  $p$ -ad rendben konvergens, ha

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = c > 0$$

Néhány megjegyzés a fenti definícióhoz:

1.  $p$  egyértelmű és  $p \geq 1$
2.  $p = 1$  esetén lineáris
3.  $p = 2$  esetén kvadratikussá,
4.  $1 < p < 2$  esetén szuperlineáris
5. A gyakorlatban az  $|x_{k+1} - x^*| \leq M|x_k - x^*|^p$  alakot használják, azt jelenti, hogy legalább  $p$ -adrendben konvergens.

**Tétel:** Tegyük fel, hogy az  $(x_k)$  sorozat konvergens,

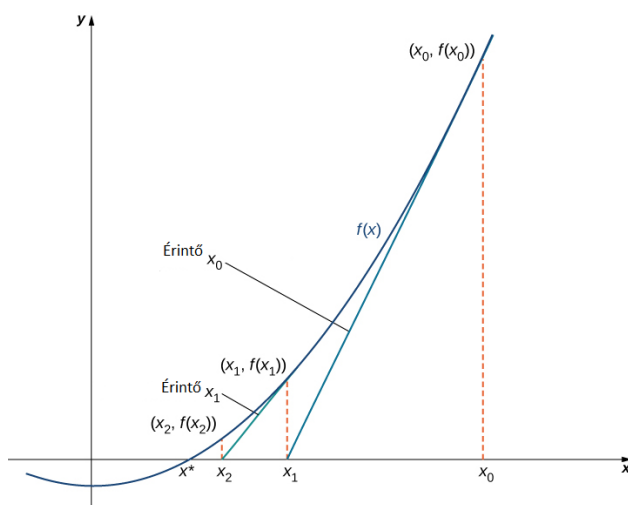
$$x_{k+1} = f(x_k) \text{ és } f'(x^*) = f''(x^*) = \dots = f^{(p-1)}(x^*) = 0, \text{ de } f^{(p)}(x^*) \neq 0.$$

Ekkor az  $(x_k)$   $p$ -adrendben konvergens.





# Newton-iteráció



1. ábra. A Newton-módszer alapötlete.

Az ábrán a legszélső,  $x_0$  pontból indulunk, felvesszük a függvény ehhez a ponthoz tartozó érintőjét, majd ennek az érintőnek a gyöke lesz a következő pont, és így tovább.

**Definíció.** Legyen  $f : [a, b] \rightarrow \mathbb{R}$ ,  $f \in D[a, b]$ , vagyis az  $f$  differenciálható az intervallumon, továbbá  $f'(x) \neq 0$ ,  $\forall x \in [a, b]$ .

Ekkor az

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n \in \mathbb{N}, \quad x_0 \in [a, b]$$

iteráció a Newton-iteráció az  $f(x) = 0$  egyenlet megoldására.

A fenti képlet a Newton-iteráció képlete. Megjegyezhető, hogy a fenti módszer is  $x_{k+1} = g(x_k)$  alakú (fixpontiteráció).

Fontos megemlíteni, hogy  $f'(x_k) = 0$  esetén nem értelmezhető a módszer. A gyakorlatban  $f'(x_k) \approx 0$  is probléma. Ha  $x^*$  többszörös gyök, akkor  $f'(x^*) = 0$ , vagyis  $x^*$  közelében  $f'(x_k)$  egyre jobban közelít 0-hoz, numerikusan instabil.

**Monoton konvergencia tétele:** Tegyük fel, hogy  $f \in \mathcal{C}^2([a, b])$  és

- $f'(x) \neq 0$ ,  $f''(x) \neq 0$ ,  $x \in [a, b]$ , valamint
- $\exists x^* \in (a, b) : f(x^*) = 0$ , és legyen
- $x_0 \in [a, b]$  kezdőpont olyan, hogy  $f(x_0)f''(x_0) > 0$  teljesüljön.

Ekkor az  $x_0$ -ból indított Newton-módszer monoton konvergál  $x^*$ -hoz.

**Lokális konvergencia tétele:** Tegyük fel, hogy  $f \in \mathcal{C}^2([a, b])$  és

- $f'(x) \neq 0$ ,  $x \in [a, b]$ , valamint
- $\exists x^* \in (a, b) : f(x^*) = 0$  és legyen

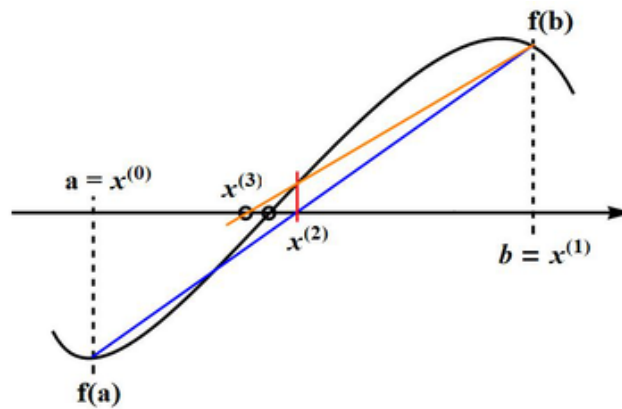
- $x_0 \in [a, b]$  kezdőpont olyan, hogy

$$|x_0 - x^*| < r := \min \left\{ \frac{1}{M}, |x^* - a|, |x^* - b| \right\}, \text{ ahol } M = \frac{\max_{[a,b]} |f''(x)|}{2 \cdot \min_{[a,b]} |f'(x)|}$$

Ekkor az  $x_0$ -ból indított Newton-iteráció konvergál az  $x^*$  megoldáshoz, és a konvergencia sebessége  $p = 2$ .

## Húrmódszer

Továbbra is  $f(x) = 0$  megoldása a cél egy adott  $[a, b]$  intervallumon. Az eljárás lényege a következő. Kezdetben  $x_0 := a, x_1 := b$ , majd meghúzzuk ezen pontok által képzett egyenest. Legyen  $x_2$  a húr gyöke. Ha  $f(x_2) = 0$ , akkor megtaláltuk a gyököt. Ha  $f(x_2) \neq 0$ , akkor folytatjuk a keresést az  $[x_0, x_2]$  vagy  $[x_2, x_1]$  intervallumban. Ha  $f(x_0)f(x_2) < 0$ , akkor  $[x_0, x_2]$  intervallumban folytatjuk, ha  $f(x_2)f(x_1) < 0$ , akkor  $[x_2, x_1]$  intervallumban. Stb.



2. ábra. Húr-módszer

Általánosan: Legyen  $x_0 := a, x_1 := b$  és  $f(a)f(b) < 0$ . Az  $(x_k, f(x_k))$  és  $(x_s, f(x_s))$  pontokon átmenő egyenesekkel közelítjük a függvényt ahol  $x_s$ -re  $f(x_s)f(x_k) < 0$  és  $s$  a legnagyobb ilyen index.  $x_{k+1}$ -et a következőképpen határozhatjuk meg:

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

**Tétel:** Legyen  $f \in \mathcal{C}^2[a, b]$  és

1.  $f(a)f(b) < 0$
2.  $M = \frac{M_2}{2m_1}$ , ahol  $0 < m_1 \leq |f'(x)|$  és  $f''(x) \leq M_2 (x \in (a, b))$
3.  $M(b - a) < 1$

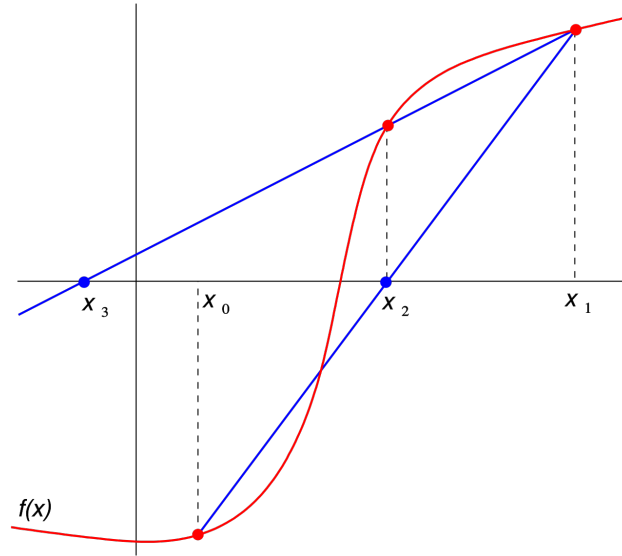
Ekkor a húrmódszer konvergens, hibabecslése pedig:

$$|x_{k+1} - x^*| \leq \frac{1}{M} (M|x_0 - x^*|)^{k+1}$$

## Szelőmódszer

A szelőmódszer lényege, hogy az  $(x_k, f(x_k))$  és  $(x_{k-1}, f(x_{k-1}))$  pontokon átmenő egyenessel közelítjük  $f$ -et, a kapott egyenes  $x$  tengellyel vett metszéspontja  $(x_{k+1})$  lesz a következő pont. Ez tulajdonképpen a húrmódszer  $s := k - 1$ -re.

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



3. ábra. A Newton-módszer alapötlete.

**Tétel:** Ha teljesülnek a Newton-módszer lokális konvergencia tételének feltételei, akkor a szelőmódszer konvergens  $p = \frac{1 + \sqrt{5}}{2}$  rendben, és hibabecslése:

$$|x_{k+1} - x^*| \leq M|x_k - x^*||x_{k-1} - x^*|$$

## Többszörös Newton-módszer

Most  $F(x) = 0$  megoldásait keressük, ahol  $F \in \mathbb{R}^n \rightarrow \mathbb{R}^n$ . Tekintsük a többszörös Newton-módszert:

$$x^{(k+1)} := x^{(k)} - [F'(x^{(k)})]^{-1} F(x^{(k)})$$

ahol

$$F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}, \quad F'(x) = \begin{bmatrix} \partial_1 f_1(x) & \partial_2 f_1(x) & \dots \\ \partial_1 f_2(x) & \partial_2 f_2(x) & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Ténylegesen az  $F'(x^{(k)})(x^{(k+1)} - x^{(k)}) = -F(x^{(k)})$  egyenletrendszert oldjuk meg.

# Interpoláció

A gyakorlatban sokszor felmerül olyan probléma, hogy egy többségében ismeretlen (csak néhány pontbeli érték ismert) vagy nagyon költségesen kiszámítható függvénnyel kellene egy megadott intervallumon dolgoznunk. Ekkor például azt tehetjük, hogy néhány pontban kiszámítjuk a függvény értékét, majd keresünk olyan egyszerűbben számítható függvényt, amelyik illeszkedik az adott pontokra. Ezután az intervallum bármely további pontjában az illesztett függvény értékeit használjuk, mint az eredeti függvény értékeinek a közelítéseit. Ilyen egyszerűbben kiszámolható függvények pl. a polinomok (polinom interpoláció).

Példa alkalmazásra: Animáció készítésénél nem szeretnénk minden egyes képkockát saját magunk elkészíteni, hanem csak bizonyos képkockákat, ún. kulcskockákat. A köztes képkockákon az egyes objektumok helyzetét szeretnénk a számítógéppel kiszámíttatani (például szeretnénk, hogy ha egy objektum egyenes vonalú, egyenletes mozgást végezne két adott pozíció között).

## Polinom interpoláció

### A polinom interpoláció feladata

Legyenek adva  $n \in \mathbb{N}$  és az  $x_k \in \mathbb{R}$ ,  $k = 0, 1, \dots, n$  különböző számok, az ún. interpolációs alappontok, valamint az  $f(x_k)$ ,  $k = 0, 1, \dots, n$  számok, az ismert függvényértékek. Keressük azt a legfeljebb  $n$ -edfokú  $p_n$  polinomot ( $p_n \in P_n$ ), amelyre:

$$p_n(x_k) = f(x_k) \quad k = 0, 1, \dots, n$$

Azaz a polinom az interpolációs alappontokban a megadott függvényértékeket veszi fel.

**Tétel:** A fenti interpolációs feladatnak létezik egyértelmű megoldása.

**Tétel:** Legyen  $n \in \mathbb{N}$  és  $x_k \in \mathbb{R}$ ,  $k = 0, 1, \dots, n$ , úgy, hogy  $x_k \neq x_l$ ,  $k \neq l$ .

Ekkor létezik olyan  $p_n(x) = \sum_{i=0}^n a_i x^i$  polinom, hogy

$$p_n(x_k) = f(x_k) \quad k = 0, 1, \dots, n$$

## Lagrange-interpoláció

Abban az esetben használjuk, ha polinommal szeretnénk közelíteni egy függvényt. Ennél a módszernél feltesszük, hogy az alappontok páronként különbözők, ami jogos, és ilyenkor adott  $x$ -re nem mehet át a függvény két  $y = f(x)$  értékekhez tartozó ponton.

A módszer egy  $n$  alappontból álló sorozat egy  $n - 1$ -edfokú polinommal közelít.

**Lagrange-alappolinomok:** Adott  $n \in \mathbb{N}$  és  $x_k \in \mathbb{R}$ ,  $k = 0, 1, \dots, n$  különböző alappontokra, ahol  $x_k \neq x_l$ ,  $k \neq l$ , a Lagrange-alappolinomokat a következőképpen definiáljuk:

$$l_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^n \frac{(x - x_j)}{(x_k - x_j)} = \frac{(x - x_0)(x - x_1) \cdots (x - x_{k-1})(x - x_{k+1}) \cdots (x - x_n)}{(x_k - x_0)(x_k - x_1) \cdots (x_k - x_{k-1})(x_k - x_{k+1}) \cdots (x_k - x_n)}$$

$k = 0, 1, \dots, n$  esetén.

Az alappontok mind  $n$ -edfokú polinomok, és a következő tulajdonsággal rendelkeznek:

$$l_k(x_j) = \begin{cases} 1 & \text{ha } j = k \\ 0 & \text{ha } j \neq k \end{cases}$$

**Tétel:** Az interpolációs feladat megoldása az alábbi polinom, amelyet az interpolációs polinom Lagrange-alakjának hívunk:

$$L_n(x) = \sum_{k=0}^n f(x_k) l_k(x)$$

A továbbiakban jelölje  $[a, b]$  az  $x_0, x_1, \dots, x_n$  alappontok által kifeszített intervallumot.

**Tétel.** Legyen  $f \in C^\infty([a, b])$  és legyen adva egy  $[a, b]$  intervallumbeli alappontrendszer sorozata:  $x_k^n$ ,  $k = 0, 1, \dots, n$ ,  $n = 0, 1, 2, \dots$

Legyen  $L_n$  az  $x_0^n, x_1^n, \dots, x_n^n$  alappontrendszerre illesztett Lagrange-interpolációs polinom ( $n = 0, 1, 2, \dots$ ).

Ha  $\exists M > 0$  úgy, hogy



$$\forall n \in \mathbb{N} : M_n \leq M^n \quad (\text{lásd előző tétel})$$

akkor az  $L_n$  sorozat egyenletesen konvergál az  $f$  függvényhez.

A Lagrange-interpoláció hátránya, hogy új alappont hozzávételével az összes számítást újra el kell végezni. Ezt kiküszöbölendő érdemesebb  $L_n$ -t a következő alakban keresni:

$$N_n(x) := A_0 + A_1(x - x_0) + \dots + A_n(x - x_0) \dots (x - x_{n-1})$$

Az együtthatók kiszámításához vezetjük be az úgynevezett osztott differenciákat.

 **Ha egy konvergens interpolációt veszünk, akkor felmerül a kérdés, hogy hogyan számítsuk ki egyre magasabb fokszámú interpolációs polinomot azért, hogy növeljük a pontosságot. Viszont a Lagrange-féle interpolációs polinomot minden lépésben újra kellene számolnunk, hiszen az új polinom hozzávételekor minden alappolinom megváltozik. Emiatt olyan előállításra van szükség, amelyik lehetővé teszi, hogy az előzőleg kiszámolt polinomot felhasználhassuk. Az alábbiakban a Newton-féle alakot tárgyaljuk, ami megoldja ezt a problémát: a polinomoknak egy rekurzív előállítását adja. Ehhez először szükség van az osztott differenciák fogalmára.** 

**Osztott differencia:** Legyenek adva az  $x_0, \dots, x_n$  különböző alappontok. Az  $f : [a, b] \rightarrow \mathbb{R}$  függvénynek a megadott alappontrendszerre vonatkozó elsőrendű osztott differenciái, ahol  $i = 1, \dots, n$

$$f[x_{i-1}, x_i] = \frac{f(x_i) - f(x_{i-1})}{x_i - x_{i-1}}$$

$n - 1$  darab másodrendű osztott differencia, ahol  $i = 2, \dots, n$

$$f[x_{i-2}, x_{i-1}, x_i] = \frac{f[x_{i-1}, x_i] - f[x_{i-2}, x_{i-1}]}{x_i - x_{i-2}}$$

ezt általánosítva a  $k$ -ad rendű osztott differencia  $i = k, \dots, n$  esetén

$$f[x_{i-k}, x_{i-k+1}, \dots, x_i] = \frac{f[x_{i-k+1}, \dots, x_i] - f[x_{i-k}, \dots, x_{i-1}]}{x_i - x_{i-k}}$$

Látható, hogy a  $k$ -adrendű osztott differencia  $k + 1$  alapontra támaszkodik.

Ha adott egy interpoláció alappontrendszer függvényértékekkel, akkor a hozzá tartozó osztott differenciákat az alábbi táblázat szerint érdemes elrendezni, és ez az elrendezés egyúttal a kiszámolást is segíti ( $k+1$  pont esetén).

$x_0$	$f(x_0)$			
		$f[x_0, x_1]$		
$x_1$	$f(x_1)$		$f[x_0, x_1, x_2]$	
		$f[x_1, x_2]$		
$x_2$	$f(x_2)$			
$\vdots$	$\vdots$	$\vdots$	$\dots$	
				$f[x_0, x_1, \dots, x_n]$
$\vdots$	$\vdots$	$\vdots$	$\dots$	
$x_{n-1}$	$f(x_{n-1})$		$f[x_{n-2}, x_{n-1}, x_n]$	
		$f[x_{n-1}, x_n]$		
$x_n$	$f(x_n)$			

**Állítás.** (Newton-interpolációs formula). A Lagrange-interpolációs polinom megadható a következő formában:

$$N_n(x) = \sum_{k=0}^n f[x_0, \dots, x_k] \prod_{j=0}^{k-1} (x - x_j)$$

Észrevehető, hogy érdekesebb a Newton-interpolációs eljárást alkalmazni, hiszen  $N_n$ -t rekurzívan is ki tudjuk fejezni az

$$N_n(x) = N_{n-1} + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

képlet segítségével, így új alappont hozzávételével gyorsabban tudunk interpolációs polinomot számolni.

## Hermite-interpoláció

Az előbbi interpolációs feladatot a következőképpen általánosíthatjuk.

Legyenek adva az egyes alappontokban a függvényértékek mellett a függvény derivált értékei is valamely rendig bezárólag. Ekkor olyan polinomot keresünk, amelyik deriváltjaival együtt illeszkedik a megadott értékekre, vagyis:

Legyenek adva  $m_0, m_1, \dots, m_n \in \mathbb{N}$  és az  $x_j \in \mathbb{R}$ ,  $j = 0, 1, \dots, n$  interpolációs alappontok, valamint az  $f^{(k)}(x_j)$   $k = 0, 1, \dots, m_j - 1$ ,  $j = 0, 1, \dots, n$  függvény- és derivált értékek.

Legyen  $m = \sum_{j=0}^n m_j$ . Keressük azt a legfeljebb  $(m - 1)$ -edfokú  $p_{m-1}$  polinomot, melyre:

$$p_{m-1}^{(k)}(x_j) = f^{(k)}(x_j) \quad (k = 0, 1, \dots, m_j - 1, \quad j = 0, 1, \dots, n)$$

**Definíció.** Az így nyert  $p$ -t Hermite-interpolációs polinomnak nevezzük és  $H_m$ -mel jelöljük.

**Tétel:** A Hermite-féle interpolációs polinom egyértelműen létezik.

**Megjegyzések:** Legyen  $j = 0, 1, \dots, n$

1. Ha  $m_j = 1$ , akkor  $m = n + 1$ . Mindegyik  $x_j$ -hez csak az  $f_j$  érték tartozik. (Lagrange-interpoláció).
2. Ha  $m_j = 2$ . Mindegyik  $x_j$ -hez az  $f_j$  érték és az  $f_j'$  derivált érték tartozik. (Hermite-Fejér-féle interpoláció). A keresett polinom pedig legfeljebb  $(2n + 1)$ -edfokú.

### A Hermite interpolációs polinom előállítása

Könnyen felírható a Newton-féle formában. Csak annyit kell tennünk, hogy kiindulunk az alappontok és a függvényértékek táblázatával és legyártjuk az osztott differenciák táblázatát. Az az egyetlen különbség most, hogy az  $x_j$  alappontot  $m_j$ -szer soroljuk fel.

Ha  $x_k$   $j$ -szer szerepel:

$$f[x_k, \dots, x_k] = \frac{f^{(j)}(x_k)}{j!}$$

**Tétel.** (Hermite-interpoláció hibája):

Legyen  $\sum_{j=0}^n m_j$ . Ha az  $f$  függvény  $m$ -szer folytonosan differenciálható, akkor

$$\forall x \in [a, b] \quad \exists \xi \in (a, b) : f(x) - H_{m-1}(x) = \frac{f^{(m)}(\xi)}{m!} \Omega_m$$

ahol  $\Omega_m(x) = (x - x_0)^{m_0} (x - x_1)^{m_1} \dots (x - x_n)^{m_n} = \prod_{i=0}^n (x - x_i)^{m_i}$

### Spline-interpoláció

A klasszikus interpoláció magas foksámra nem előnyös alkalmazni. Sok esetben előnyösebb a szakaszonként (alacsony) adott foksámú interpoláció (spline interpoláció) alkalmazása.

💡 Bizonyos esetekben probléma lehet a Lagrange-interpolációval, ahol a polinom foksáma nő az alappontok számával. Ennek következményeként indokolatlanul nagy hullámzások jelennek meg a polinom grafikus képében, két egymást követő pont között. Ez annak tudható be, hogy az eredeti alappontokat szolgáltató  $f(x)$  függvény nem polinomiális, és egy közelítő polinom csak úgy tud eleget tenni a feltételeknek, hogy közben lokális minimum és maximum helyeken halad át. Erre nyújt megoldást a spline interpoláció.

A lényeg, hogy a kicsi foksámú polinomokból rakjuk össze a közelítő polinomot, mégpedig úgy, hogy először keresünk egy polinomot az első két alappontra, majd a második és harmadik alappontra (és így tovább), de mindezt úgy tesszük, hogy közben figyelünk az illeszkedési pontokban a derivált értékekre (ezzel biztosítva a görbék tökéletes illeszkedését, mivel a spline-ok akkor illeszkednek szépen, ha az alappontokban nem csak az első, hanem a második derivált értéke is megegyezik).

Létezik elsőfokú, másodfokú, harmadfokú és néhány magasabb foksámú spline, de gyakorlatban a harmadfokú (cubic) spline-ok a legelterjedtebbek, mert amikor már megvan egy szakaszon a közelítés, és oda akarunk csatlakoztatni egy következő spline-t, akkor van két adatponton érték, illetve a csatlakozási ponton az első és második derivált. Ez négy feltétel, amire egy harmadfokú polinomot lehet illeszteni. <💡

## $l$ -edfokú spline

Legyen  $\Omega_n := [a = x_0 < \dots < x_n < b]$  felosztás, ahol  $l_k := [x_{k-1}, x_k]$  részintervallum ( $k = 1, \dots, n$ ). Az  $\mathbf{S}_\ell : [a, b] \rightarrow \mathbb{R}$  függvényt egy  $\ell$ -edfokú spline-nak nevezzük, ha:

1.  $\mathbf{S}_\ell|_{l_k} \in P_\ell \quad (k = 1, \dots, n)$
2.  $\mathbf{S} \in \mathcal{C}^{\ell-1}[a, b]$  (a teljes intervallumon  $(\ell - 1)$ -szer folytonosan deriválható)

Részintervallumonkénti polinomok együtthatóinak megkeresése:

$$p_k(x) = \sum_{j=0}^{\ell} a_j^{(k)} \cdot (x - x_{k-1})^j \quad (x \in l_k)$$

### Spline megadás intervallumonként

#### $\ell = 1$ elsőfokú spline megadása

A legegyszerűbb eset a szakaszonként lineáris spline interpoláció. Ebben az esetben az interpolációban szakaszonként egy egyenessel kötünk össze két egymás mellett lévő pontot.

$$p_k(x) = a_1^{(k)}(x - x_{k-1}) + a_0^{(k)} \quad (x \in l_k)$$

A megoldáshoz írjuk fel az interpolációs feltételeket az  $l_k$  részintervallum két szélére.

$$\begin{aligned} p_k(x_{k-1}) &= a_1^{(k)}(x_{k-1} - x_{k-1}) + a_0^{(k)} = f(x_{k-1}) \\ p_k(x_k) &= a_1^{(k)}(x_k - x_{k-1}) + a_0^{(k)} = f(x_k) \\ &\Downarrow \\ a_1^{(k)} &= \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = f[x_{k-1}, x_k] \end{aligned}$$

Az interpolációs feltételből a folytonosság azonnal következik.

#### $\ell = 2$ másodfokú (kvadrátikus) spline megadása

$$p_k(x) = a_2^{(k)}(x - x_{k-1})^2 + a_1^{(k)}(x - x_{k-1}) + a_0^{(k)} \quad (x \in l_k)$$

- Az ismeretlenek száma:  $3n$ .
- A feltételek száma:
  - interpolációs feltétel minden részintervallum két szélére:  **$2n$**  feltétel ( $n$  intervallumra felosztva, intervallumonként 2 végponttal)
- minden belső osztóponttra a folytonos differenciálhatóság:  **$n-1$**  feltétel

Összesen  $2n + n - 1 = \mathbf{3n-1}$  feltétel van, vagyis az egyértelműséghez hiányzik egy feltétel. Ezt peremfeltételként szokás megadni:

$$\mathbf{S}'(a) = f'(a), \quad \mathbf{S}'(b) = f'(b) \text{ (Hermite feltételek)}$$

A feladat szétbontható  $n$  db Hermite-interpolációs feladattá.



## $\ell = 3$ harmadfokú (kübös) spline megadása

$$p_k(x) = a_3^{(k)}(x - x_{k-1})^3 + a_2^{(k)}(x - x_{k-1})^2 + a_1^{(k)}(x - x_{k-1}) + a_0^{(k)} \quad (x \in I_k)$$

- Az ismeretlenek száma:  $4n$ .
  - interpolációs feltétel minden részintervallum két szélére:  **$2n$**  feltétel  
( $n$  intervallumra felosztva, intervallumonként 2 végponttal)
- minden belső osztóponttra  $\mathbf{S}'_3 \in \mathcal{C}$ :  **$n-1$**  feltétel
- minden belső osztóponttra  $\mathbf{S}''_3 \in \mathcal{C}$ :  **$n-1$**  feltétel

Összesen  $2n + n - 1 + n - 1 = 4n - 2$  feltétel van, vagyis az egyértelműséghez hiányzik két feltétel. Ezt peremfeltételként szokás megadni:

$$\mathbf{S}'(a) = f'(a), \quad \mathbf{S}'(b) = f'(b) \quad (\text{Hermite feltételek})$$

$$\mathbf{S}''(a) = \mathbf{S}''(b) = 0 \quad (\text{Természetes peremfeltételek})$$

Csak periodikus függvények közelítése esetén, ha  $[a, b]$  a periódus többszöröse. Ekkor  $f(a) = f(b)$  és a két hiányzó feltétel:

$$\mathbf{S}'(a) = \mathbf{S}'(b), \text{ és } \mathbf{S}''(a) = \mathbf{S}''(b) \quad (\text{Periodikus feltételek})$$

## B-Spline

**Definíció.** (Függvény tartója): Az  $f : \mathbb{R} \rightarrow \mathbb{R}$  **függvény tartója** a következő valós számhalmaz:

$$\text{supp}(f) := \overline{\{x \in \mathbb{R} : f(x) \neq 0\}}$$

**Definíció.** (Alappontrendszer):  $\Omega_\infty := \{\dots, x_{-n}, \dots, x_{-1}, x_0, x_1, \dots, x_n, \dots\}$

$\mathbf{S}_l(\Omega_\infty)$  az  $\Omega_n$  alappontrendszeren értelmezett  $l$ -edfokú spline-ok halmaza.

**B-spline:** A  $B_{l,k} \in \mathbf{S}_l(\Omega_\infty)$ , ( $k \in \mathbb{Z}$ )  $l$ -edfokú spline függvények rendszerét B-spline függvényeknek nevezzük, ha az alábbi feltételek teljesülnek:

1.  $B_{l,k}(x) \geq 0$
2.  $\text{supp}(B_{l,k}) = [x_k, x_{k+l+1}]$ , azaz a tartója minimális
3.  $\sum_{k \in \mathbb{Z}} B_{l,k}(x) \equiv 1 \quad (\forall x \in \mathbb{R})$

## Legkisebb négyzetek módszere

Gyakorlati feladatok során adódik a következő probléma. Egy elsőfokú függvényt mérünk bizonyos pontokban, de a mérési hibák miatt ezek nem lesznek egy egyenesen. Ekkor olyan egyenest keresünk, amelyik az alábbi értelemben legjobban illeszkedik a megadott mérési ponthalmazra.

## Egyenes eset

Legyenek adva az  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  mérési pontok.

Keressük azt a  $p_1(x) = a + bx$  legfeljebb elsőfokú polinomot, amelyre a következő kifejezés minimális.

$$\sum_{i=1}^n (y_i - p_1(x_i))^2 = \sum_{i=1}^n (y_i - a - bx_i)^2$$

Ez azt jelenti, hogy azt az egyenes keressük, amelyre a függvényértékek hibáinak négyzetösszege minimális.

## Polinom eset

Adottak az  $n, N \in \mathbb{N}$ , ahol  $n \ll N$  és  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$  mérési pontok, ahol az  $x_i$  alappontok különbözők.

Keressük azt a  $p_n(x) = a_0 + a_1x + \dots + a_nx^n = \sum_{i=0}^n a_jx^j$  legfeljebb  $n$ -edfokú polinomot, melyre a következő kifejezés minimális.

$$\sum_{i=1}^m (y_i - p_n(x_i))^2$$

**Definíció:** Az  $m \times n$ -es  $A$  mátrix transzponáltján azt az  $A^T$ -vel jelölt  $n \times m$ -es mátrixot értjük, amelyet az  $A$  sorainak és oszlopainak felcserélésével kapunk. Azaz

$$A^T = [a_{ij}]^T = [a_{ji}]$$

A feladat megoldásához tekintsük annak egy átfogalmazását.

Vegyük a  $p_n(x_i) = y_i$ ,  $(i = 1, 2, \dots, m)$  egyenletrendszert. Ez a rendszer az ismeretlen  $a_i$  együtthatókra nézve lineáris, mégpedig túlhatározott, amelynek az  $A$  mátrixa egy téglalap alakú Vandermonde-mátrix  $A \in \mathbb{R}^{m \times (n+1)}$ , a  $b \in \mathbb{R}^m$  jobb oldali vektorra pedig a függvényértékekből adódik:

$$\begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^n \\ 1 & x_2 & x_2^2 & \dots & x_2^n \\ 1 & x_3 & x_3^2 & \dots & x_3^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_m & x_m^2 & \dots & x_m^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

Ezen jelölésekkel a minimalizálandó kifejezés  $\|Az - b\|_2^2$ , ahol  $z = [a_0, a_1, \dots, a_n]^T$  a keresett együtthatók vektorra. A feladat megoldását a Gauss-féle normálegyenletek adják:

$$A^T A z = A^T b$$

A fenti lineáris egyenletrendszert kell megoldani  $z$ -re.

**n=1 eset:** Ekkor a feladatot gyakran lineáris regresszióknak is hívjuk. Ebben az esetben

$$A^T A = \begin{bmatrix} m & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i & \sum_{i=1}^m x_i^2 \end{bmatrix}, A^T b = \begin{bmatrix} \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i y_i \end{bmatrix}, z = \begin{bmatrix} b \\ a \end{bmatrix}$$

## Kiegészítés

### Elégséges feltétel a konvergenciára (lineáris egyenletrendszer)

Tekintsük az  $\underline{x} = B\underline{x} + \underline{r}$  egyenletet, ahol  $\|B\| < 1$  valamely indukált vagy illeszkedő normában.

Ekkor

- $\exists! x^* \in \mathbb{R}^n : x^* = Bx^* + \underline{r}$
- $\forall \underline{x}^{(0)} \in \mathbb{R}^n$  kezdőérték esetén, az  $\underline{x}^{(k+1)} = B\underline{x}^{(k)} + \underline{r}$  rekurzióval definiált sorozat konvergens, és a határértéke az  $x^*$  lesz, vagyis

$$\lim_{k \rightarrow \infty} \underline{x}^{(k)} = \underline{x}^*$$

- teljesül az alábbi hibabecslés is:

$$\|\underline{x}^{(k)} - \underline{x}^*\| \leq \frac{\|B\|^k}{1 - \|B\|} \|\underline{x}^{(1)} - \underline{x}^{(0)}\|$$

---

Az  $L$  mátrix *alsó háromszögmátrix*, az  $A$  mátrix szigorú alsó része, vagyis

$$l_{i,j} = a_{i,j}, \text{ ha } i > j, \text{ különben } 0, \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{x} & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{x} & \cdots & \mathbf{x} & 0 \end{pmatrix}$$

az  $U$  *felső háromszögmátrix*, az  $A$  mátrix szigorú felső része, vagyis

$$u_{i,j} = a_{i,j}, \text{ ha } i < j, \text{ különben } 0, \begin{pmatrix} 0 & \mathbf{x} & \cdots & \mathbf{x} \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{x} \\ 0 & \cdots & 0 & 0 \end{pmatrix}$$

a  $D$  pedig *invertálható diagonális mátrix* az  $A$  mátrix diagonális része, vagyis

$$d_{i,j} = a_{i,j}, \text{ ha } i = j, \text{ különben } 0, \begin{pmatrix} \mathbf{x} & 0 & \cdots & 0 \\ 0 & \mathbf{x} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathbf{x} \end{pmatrix}$$

---

A Jacobi-iteráció kanonikus alakját a következő átalakításokkal kaphatjuk meg:

$$\begin{aligned} \underline{x}^{(k+1)} &= -D^{-1}(L + U)\underline{x}^{(k)} + D^{-1}\underline{b} \\ D\underline{x}^{(k+1)} + (L + U)\underline{x}^{(k)} &= \underline{b} \\ D(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + (L + U)\underline{x}^{(k)} + D\underline{x}^{(k)} &= \underline{b} \\ D(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + \underbrace{(L + U + D)\underline{x}^{(k)}}_{=A} &= \underline{b} \\ D(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + A\underline{x}^{(k)} &= \underline{b} \end{aligned}$$

**Állítás.** Ha a Jacobi iteráció által előállított  $(\underline{x}^{(k)})$  vektorsorozat konvergens, azaz létezik  $\underline{x}^*$ , melyre  $\lim_{k \rightarrow \infty} \underline{x}^{(k)} = \underline{x}^*$ , akkor  $\underline{x}^*$  megoldása az  $A\underline{x} = \underline{b}$  egyenletrendszernek.

**Bizonyítás.** A Jacobi-iteráció kanonikus képletében térjünk át limeszre, és felhasználva, hogy  $\lim \lambda x = \lambda \lim x$  és  $\lim(A + B) = \lim A + \lim B$

$$\lim_{k \rightarrow \infty} \left[ D(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + A\underline{x}^{(k)} \right] = D \lim_{k \rightarrow \infty} \underbrace{(\underline{x}^{(k+1)} - \underline{x}^{(k)})}_{\rightarrow \underline{x}^* - \underline{x}^* = 0} + A \lim_{k \rightarrow \infty} \underbrace{\underline{x}^{(k)}}_{\rightarrow \underline{x}^*} = A\underline{x}^* = \underline{b}$$

Felhasználva, hogy  $\lim \lambda x = \lambda \lim x$  és  $\lim(A + B) = \lim A + \lim B$ .

Ha az  $Ax = b$  lineáris egyenletrendszer mátrixa szigorúan diagonálisan domináns

$$\|B_J\|_\infty = \| -D^{-1}(L + U) \|_\infty = \max_{1 \leq k \leq n} \sum_{\substack{i=1 \\ i \neq k}}^n \frac{|a_{i,k}|}{|a_{i,i}|} = \max_{1 \leq k \leq n} \frac{1}{|a_{i,i}|} \sum_{\substack{i=1 \\ i \neq k}}^n |a_{i,k}| < 1 \quad (\text{soraira})$$

$$\|B_J\|_1 = \| -D^{-1}(L + U) \|_1 = \max_{1 \leq i \leq n} \sum_{\substack{k=1 \\ k \neq i}}^n \frac{|a_{i,k}|}{|a_{i,i}|} = \max_{1 \leq i \leq n} \frac{1}{|a_{i,i}|} \sum_{\substack{k=1 \\ k \neq i}}^n |a_{i,k}| < 1 \quad (\text{oszlopaira})$$

akkor a Jacobi-iteráció konvergens.

Példa Jacobi-iterációra. Tekintsük a következő egyenletrendszert:

$$\begin{bmatrix} 5 & -2 & 1 \\ 0 & 3 & -1 \\ 2 & -1 & 6 \end{bmatrix}_A \underline{x} = \begin{bmatrix} 4 \\ 2 \\ 7 \end{bmatrix}_b$$

Képezzük a Jacobi iteráció komponensenkénti alakját, majd számoljuk ki az első négy iterációs lépést az  $x^{(0)} = [0, 0, 0]^T$  vektorból kiindulva.

$$5x_1 - 2x_2 + x_3 = 4$$

$$3x_2 - x_3 = 2$$

$$2x_1 - x_2 + 6x_3 = 7$$

↓

$$5x_1 = 4 + 2x_2 - x_3$$

$$3x_2 = 2 + x_3$$

$$6x_3 = 7 - 2x_1 + x_2$$

↓

$$x_1 = \frac{4}{5} + \frac{2x_2}{5} - \frac{x_3}{5}$$

$$x_2 = \frac{2}{3} + \frac{x_3}{3}$$

$$x_3 = \frac{7}{6} - \frac{2x_1}{6} + \frac{x_2}{6}$$

↓ (átrendezés után)

$$x_1^{(k+1)} = \frac{2}{5}x_2^{(k)} - \frac{1}{5}x_3^{(k)} + \frac{4}{5}$$

$$x_2^{(k+1)} = \frac{1}{3}x_3^{(k)} + \frac{2}{3}$$

$$x_3^{(k+1)} = -\frac{2}{6}x_1^{(k)} + \frac{1}{6}x_2^{(k)} + \frac{7}{6}$$

Írjuk vissza mátrix alakra:

$$\begin{bmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{bmatrix} = \begin{bmatrix} 0 & \frac{2}{5} & -\frac{1}{5} \\ 0 & 0 & \frac{1}{3} \\ -\frac{2}{6} & \frac{1}{6} & 0 \end{bmatrix}_A \cdot \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \\ x_3^{(k)} \end{bmatrix} + \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}$$

A kezdővektor  $x^{(0)} = [0, 0, 0]^T$ , így az első négy iterációs lépés a következő.

$$\begin{aligned}
 x^{(1)} &= \begin{bmatrix} 0 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 0 & \frac{1}{3} \\ -\frac{2}{6} & \frac{1}{6} & 0 \end{bmatrix}_A \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{x^{(0)}} + \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_b = \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_{x^{(1)}} \\
 x^{(2)} &= \begin{bmatrix} 0 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 0 & \frac{1}{3} \\ -\frac{2}{6} & \frac{1}{6} & 0 \end{bmatrix}_A \cdot \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_{x^{(1)}} + \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_b = \begin{bmatrix} \frac{5}{8} \\ \frac{19}{18} \\ \frac{91}{90} \end{bmatrix}_{x^{(2)}} \\
 x^{(3)} &= \begin{bmatrix} 0 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 0 & \frac{1}{3} \\ -\frac{2}{6} & \frac{1}{6} & 0 \end{bmatrix}_A \cdot \begin{bmatrix} \frac{5}{8} \\ \frac{19}{18} \\ \frac{91}{90} \end{bmatrix}_{x^{(2)}} + \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_b = \begin{bmatrix} \frac{51}{50} \\ \frac{271}{270} \\ \frac{115}{108} \end{bmatrix}_{x^{(3)}} \\
 x^{(4)} &= \begin{bmatrix} 0 & -\frac{2}{5} & \frac{1}{5} \\ 0 & 0 & \frac{1}{3} \\ -\frac{2}{6} & \frac{1}{6} & 0 \end{bmatrix}_A \cdot \begin{bmatrix} \frac{51}{50} \\ \frac{271}{270} \\ \frac{115}{108} \end{bmatrix}_{x^{(3)}} + \begin{bmatrix} \frac{4}{5} \\ \frac{2}{3} \\ \frac{7}{6} \end{bmatrix}_b = \begin{bmatrix} \frac{2669}{2700} \\ \frac{331}{324} \\ \frac{8051}{8100} \end{bmatrix}_{x^{(4)}}
 \end{aligned}$$

A Gauss-Seidel kanonikus alakját a következő átalakításokkal kaphatjuk meg:

$$\begin{aligned}
 \underline{x}^{(k+1)} &= -(D + L)^{-1}U\underline{x}^{(k)} + (D + L)^{-1}\underline{b} \\
 (D + L)\underline{x}^{(k+1)} + U\underline{x}^{(k)} &= \underline{b} \\
 (D + L)\underline{x}^{(k+1)} - (D + L)\underline{x}^{(k)} + U\underline{x}^{(k)} + (D + L)\underline{x}^{(k)} &= \underline{b} \\
 (D + L)(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + \underbrace{(U + D + L)\underline{x}^{(k)}}_{=A} &= \underline{b} \\
 (D + L)(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + A\underline{x}^{(k)} &= \underline{b}
 \end{aligned}$$

**Állítás.** Ha a Gauss-Seidel iteráció által előállított  $(\underline{x}^{(k)})$  vektorsorozat konvergens, azaz létezik  $\underline{x}^*$ , melyre  $\lim_{k \rightarrow \infty} \underline{x}^{(k)} = \underline{x}^*$ , akkor  $\underline{x}^*$  megoldása az  $A\underline{x} = \underline{b}$  egyenletrendszernek.

**Bizonyítás.** A Gauss-Seidel-iteráció kanonikus képletében térjünk át limeszre, és felhasználva, hogy  $\lim \lambda x = \lambda \lim x$  és  $\lim(A + B) = \lim A + \lim B$ .

$$\lim_{k \rightarrow \infty} \left[ (D + L)(\underline{x}^{(k+1)} - \underline{x}^{(k)}) + A\underline{x}^{(k)} \right] = (D + L) \lim_{k \rightarrow \infty} \underbrace{(\underline{x}^{(k+1)} - \underline{x}^{(k)})}_{\rightarrow \underline{x}^* - \underline{x}^* = 0} + A \lim_{k \rightarrow \infty} \underbrace{\underline{x}^{(k)}}_{\rightarrow \underline{x}^*} = A\underline{x}^* = \underline{b}$$

Példa a Gauss-Seidel iterációra. Tekintsük a következő egyenletrendszert:

$$\begin{bmatrix} -5 & -2 & 1 \\ 1 & 3 & -1 \\ 2 & -1 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} -5_{a_{1,1}} & -2_{a_{1,2}} & 1_{a_{1,3}} \\ 1_{a_{2,1}} & 3_{a_{2,2}} & -1_{a_{2,3}} \\ 2_{a_{3,1}} & -1_{a_{3,2}} & 6_{a_{3,3}} \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 = 1 \\ b_2 = 1 \\ b_3 = 1 \end{bmatrix}$$

A kezdővektor  $x^{(0)} = [0, 0, 0]^T$ , így az első két iterációs lépés a következő.

$$\begin{aligned} x_1^{(1)} &= -\frac{1}{a_{1,1}} \left( \sum_{j=1}^0 a_{1,j} x_j^{(1)} + \sum_{j=2}^3 a_{1,j} x_j^{(0)} - b_1 \right) = -\frac{1}{-5} \left( 0 + a_{1,1} x_1^{(0)} + a_{1,3} x_3^{(0)} - b_1 \right) = \\ &= \frac{1}{5} \left( -5x_1^{(0)} + x_3^{(0)} - 1 \right) = -x_1^{(0)} + \frac{1}{5} x_3^{(0)} - \frac{1}{5} = -0 + \frac{1}{5} \cdot 0 - \frac{1}{5} = -\frac{1}{5} \end{aligned}$$

$$\begin{aligned} x_2^{(1)} &= -\frac{1}{a_{2,2}} \left( \sum_{j=1}^1 a_{2,j} x_j^{(1)} + \sum_{j=3}^3 a_{2,j} x_j^{(0)} - b_2 \right) = -\frac{1}{3} \left( 1 \cdot x_1^{(1)} + a_{2,3} x_3^{(0)} - b_2 \right) = \\ &= -\frac{1}{3} \left( -\frac{1}{5} - 0 - 1 \right) = -\frac{5}{15} \left( -\frac{3}{15} - \frac{15}{15} \right) = \frac{90}{225} = \frac{2}{5} \end{aligned}$$

$$\begin{aligned} x_3^{(1)} &= -\frac{1}{a_{3,3}} \left( \sum_{j=1}^2 a_{3,j} x_j^{(1)} + \sum_{j=4}^3 a_{3,j} x_j^{(0)} - b_3 \right) = -\frac{1}{6} \left( 2 \cdot x_1^{(1)} - 1 \cdot x_2^{(1)} + 0 - 1 \right) = \\ &= -\frac{1}{6} \left( 2 \cdot -\frac{1}{5} - 1 \cdot \frac{2}{5} - 1 \right) = -\frac{1}{6} \left( -\frac{2}{5} - \frac{2}{5} - \frac{5}{5} \right) = \frac{3}{10} \end{aligned}$$

Tehát  $x^{(1)} = \left[-\frac{1}{5}, \frac{2}{5}, \frac{3}{10}\right]^T$ . Az  $x^{(2)}$  pedig hasonlóan számítható:  $x^{(2)} = \left[-\frac{3}{10}, \frac{8}{18}, \frac{16}{45}\right]^T$ .

*Példa:* Adjuk meg azt a legfeljebb harmadfokú polinomot, amelyik illeszkedik az alábbi pontokra:

$$(x_0 = -1, y_0 = 1), (x_1 = 0, y_1 = 0), (x_2 = 1, y_2 = -1), (x_3 = 2, y_3 = 4)$$

Az alappontok táblázatba rendezve:

$x$	$-1$	$0$	$1$	$2$
$p_3(x)$	$1$	$0$	$-1$	$4$

Írjuk fel a Lagrange-alakot, amihez előbb felírjuk a következő négy alappolinomot

$$l_0(x) = \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} = \frac{(x - 0)(x - 1)(x - 2)}{(-1 - 0) \cdot (-1 - 1) \cdot (-1 - 2)} = \frac{x(x - 1)(x - 2)}{(-1) \cdot (-2) \cdot (-3)}$$

$$l_1(x) = \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} = \frac{(x - (-1))(x - 1)(x - 2)}{(0 - (-1)) \cdot (0 - 1) \cdot (0 - 2)} = \frac{(x + 1)(x - 1)(x - 2)}{1 \cdot (-1) \cdot (-2)}$$

$$l_2(x) = \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} = \frac{(x - (-1))(x - 0)(x - 2)}{(1 - (-1)) \cdot (1 - 0) \cdot (1 - 2)} = \frac{(x + 1)x(x - 2)}{2 \cdot 1 \cdot (-1)}$$

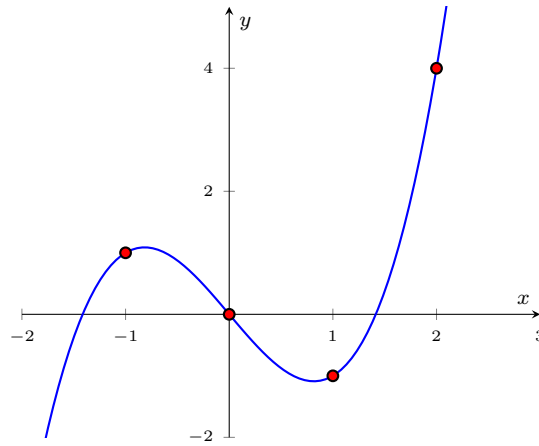
$$l_3(x) = \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} = \frac{(x - (-1))(x - 0)(x - 1)}{(2 - (-1)) \cdot (2 - 0) \cdot (2 - 1)} = \frac{(x + 1)x(x - 1)}{3 \cdot 2 \cdot 1}$$

Ezek után képezzük a Lagrange-alakot, ami

$$L_3(x) = f(x_0)l_0(x) + f(x_1)l_1(x) + f(x_2)l_2(x) + f(x_3)l_3(x) = 1 \cdot l_0(x) + 0 \cdot l_1(x) + (-1) \cdot l_2(x) + 4 \cdot l_3(x)$$

Az egyszerűsítés elvégzése után azt kapjuk, hogy

$$L_3(x) = x^3 - 2x$$

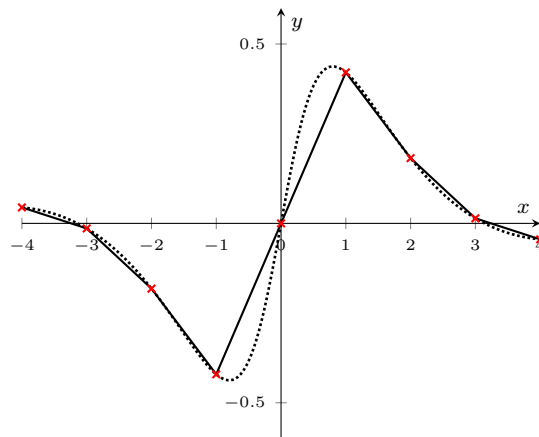



---

Példa:  $\frac{\sin(x)}{1+x^2}$ . (lineáris spline)

Az  $x_k$  alappontok a következők:

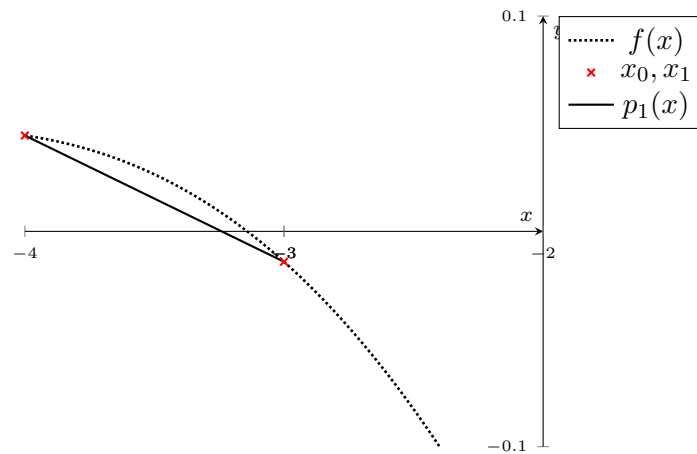
$x_0 = -4; x_1 = -3; x_2 = -2; x_3 = -1; x_4 = 0; x_5 = 1; x_6 = 2; x_7 = 3; x_8 = 4$ .



Látható, hogy a lineáris spline jól közelíti a függvényt az egyes intervallumokon, kivéve a  $[-1, 1]$ -es intervallumot. Itt a függvény nem képes az eredeti görbületre illeszkedni.

$p_1$  polinom az  $[x_0, x_1]$  intervallumon.

$$\begin{aligned}
 p_1(x) &= a_1^{(1)}(x - x_0) + a_0^{(1)} \\
 \mathbf{a}_0 &= f(-4) = -\frac{\sin(4)}{17} \\
 \mathbf{a}_1 &= \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} = \frac{10\sin(4) - 17\sin(3)}{170} \\
 p_1(x) &= \frac{10\sin(4) - 17\sin(3)}{170}x + \frac{4(10\sin(4) - 17\sin(3))}{170} - \frac{\sin(4)}{17} = \dots = \\
 &= \frac{1}{170}((10\sin(4) - 17\sin(3))x + 30\sin(4) - 68\sin(3))
 \end{aligned}$$



Példa: Számoljuk ki, majd rajzoljuk fel az alábbi pontokra négyzetesen legjobban illeszkedő egyenest:  $(x_1 = -1; y_1 = 0)$   $(x_2 = 0; y_2 = 1)$   $(x_3 = 1; y_3 = 0)$   $(x_4 = 2; y_4 = 2)$  (legkisebb négyzetek módszere)

$$A^T A = \begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ 2 & 6 \end{bmatrix}$$

$$A^T b = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$A^T A \underline{a} = A^T \underline{b}$$

$$\begin{bmatrix} 4 & 2 \\ 2 & 6 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$\begin{aligned} 4a_0 + 2a_1 &= 3 \Rightarrow 4a_0 = 3 - 2a_1 \Rightarrow a_0 = \frac{3 - 2a_1}{4} \\ 2a_0 + 6a_1 &= 4 \Rightarrow 2 \cdot \frac{3 - 2a_1}{4} + 6a_1 = 4 \Rightarrow 3 - 2a_1 + 12a_1 = 8 \Rightarrow 10a_1 = 5 \Rightarrow a_1 = \frac{1}{2} \\ a_0 &= \frac{3 - 2 \cdot \frac{1}{2}}{4} = \frac{1}{2} \end{aligned}$$

A megoldás:  $l(x) = \frac{1}{2}x + \frac{1}{2}$

