

# Zenbakizko Analisia



José Ignacio Royo Prieto



## ZENBAKIZKO ANALISIA

**José Ignacio Royo Prieto**

Apunte-bilduma hau ez da interpretatu behar ohiko testu-liburutzat, baizik eta klaseak jarraitzeko txostentzat. Esate baterako, txosten honetan aurkituko dituzu ikasgaiko edukin guztiak, teorema guztiak ondo enuntziatuta, frogapenak, etabar. Aldiz, ez dituzu aurkituko adibide guztiak, ezta ariketen garapenik ere: ariketak eta adibideak klasean garatuko ditugu, arbelean.

Txosten honen egitura, edukinak eta adibide batzuk gogorki oinarrituta daude Ana Rosa Carrera eta José Ramón Alonso irakasleen “Análisis Numérico, Notas de Clase” liburuan (Eskolako bibliotekan eskuragarri eta argitalpen zerbitzuan salgai).

**Bilbon, 2010eko otsailak 12**



*La literatura, la ciencia y el arte deben de ser servidas por voluntarios. Sólo con esa condición conseguirán liberarse del yugo del estado, del capital y de la medianía burguesa que los ahogan.*

...

*Ciencia e industria, saber y aplicación, descubrimiento y realización práctica que conduce a nuevas invenciones, trabajo cerebral y trabajo manual, idea y labor de los brazos, todo se enlaza. Cada descubrimiento, cada progreso, cada aumento de la riqueza de la humanidad tiene su origen en el conjunto del trabajo manual y cerebral, pasado y presente. Entonces, ¿qué derecho asiste a nadie para apropiarse la menor partícula de ese inmenso todo y decir: Esto es mío y no vuestro?*

(PIOTR KROPOTKIN, *La conquista del Pan*)

*God save us! We're in the hands of engineers!*

(IAN MALCOLM, *Jurassic Park*)



# Gaien Aurkibidea

<b>1</b>	<b>Errorea eta algoritmoen ezaugarriak</b>	<b>1</b>
1.1	Errorearen jatorriak . . . . .	1
1.1.1	Motze-errorea . . . . .	1
1.1.2	Borobiltze errorea . . . . .	2
1.1.3	Errore totala . . . . .	2
1.1.4	Hasierako errorea . . . . .	2
1.2	Errore absolutu eta erlatiboa . . . . .	3
1.3	Koma flotatzaileko adierazpena . . . . .	4
1.3.1	Borobiltze-prozesua . . . . .	4
1.3.2	Ordenagailuaren zehaztasuna . . . . .	5
1.4	Koma flotatzaileko aritmetika . . . . .	6
1.4.1	Batuketa ez da elkarkorra . . . . .	7
1.4.2	Elementu neutroa ez da bakarra . . . . .	7
1.4.3	Deuseztapen efektua . . . . .	8
1.5	Algoritmoen propietateak . . . . .	10
1.5.1	Konbergentzia . . . . .	10
1.5.2	Egonkortasuna eta problemaren baldintzatzea . . . . .	11
1.5.3	Eragiketa-kostua . . . . .	12
1.5.4	Memoriaren erabilera . . . . .	14
1.5.5	Metodo motak . . . . .	14
<b>2</b>	<b>Ekuzio linealezko sistemen ebazpena</b>	<b>15</b>
2.1	Sarrera . . . . .	15
2.2	Atzerakako ordezkapena . . . . .	18
2.3	Ezabapen Gaussiarra . . . . .	19
2.3.1	Pibotaketarik gabeko Gauss metodoa . . . . .	19
2.3.2	Pibotaketarik gabeko Gauss-Jordan metodoa . . . . .	23

2.3.3	Pibotaketa teknikak . . . . .	25
2.4	Ezabapen metodo trinkoak . . . . .	29
2.4.1	Sarrera . . . . .	29
2.4.2	Aurrerakako ordezkapena . . . . .	31
2.4.3	Doolittle eta Crout algoritmoak . . . . .	31
2.4.4	Cholesky metodoa . . . . .	35
2.4.5	Jacobiren matrize tridiagonalak . . . . .	36
2.5	Errorea eta Arazoaren Baldintzaketa . . . . .	38
2.5.1	Norma bektorialak eta matrizialak. Norma naturalak . . . . .	38
2.5.2	Errorearen azterketa . . . . .	41
2.6	Iteraziozko Teknikak . . . . .	44
2.6.1	Sarrera . . . . .	44
2.6.2	Jacobi metodoa . . . . .	44
2.6.3	Gauss-Seidel metodoa . . . . .	46
2.6.4	Iteraziozko metodoen konbergentzia . . . . .	46
2.6.5	Lasaikuntza Teknikak . . . . .	48
<b>3</b>	<b>Ekuzio ez linealen ebazpen hurbildua</b>	<b>51</b>
3.1	Sarrera . . . . .	51
3.2	Metodo Itxiak . . . . .	52
3.2.1	Bisekzio metodoa (Bolzano metodoa) . . . . .	52
3.2.2	Lekugune faltsuaren metodoa ( <i>Regula falsi</i> ) . . . . .	54
3.3	Metodo Irekiak . . . . .	55
3.3.1	Puntu Finkoko Iterazioa . . . . .	55
3.4	Iteraziozko metodoen konbergentzia . . . . .	59
3.4.1	Puntu finko metodoen konbergentzia . . . . .	60
3.4.2	Bisekzio eta <i>Regula Falsi</i> metodoen konbergentzia . . . . .	61
3.5	Konbergentzia Arintzeko Teknikak . . . . .	62
3.5.1	Aitken $\Delta^2$ algoritmoa . . . . .	62
3.5.2	Steffensen algoritmoa . . . . .	63
3.6	Newton-Raphson Metodoa . . . . .	64
3.7	Ebakitzaileko metodoa . . . . .	66
3.8	Erro anitzkoitzak . . . . .	67
<b>4</b>	<b>Autobalio eta autobektoreen kalkulua</b>	<b>71</b>
4.1	Sarrera . . . . .	71
4.2	Berreturako metodoa . . . . .	72
4.2.1	Autobalio menperatzailearen kalkulua . . . . .	72
4.2.2	Modulo txikieneko autobalioaren kalkulua . . . . .	74
4.2.3	Berretura alderantzizkoko metodo lerratua . . . . .	75



<b>5</b>	<b>Interpolazioa</b>	<b>77</b>
5.1	Sarrera . . . . .	77
5.2	Polinomio interpolatzailea. Lagrange metodoa . . . . .	77
5.3	Newtonen diferentzia zatituen metodoa . . . . .	79
5.3.1	Diferentzia zatituen propietateak . . . . .	80
5.3.2	Diferentzia zatituak kalkulatzeko algoritmoa . . . . .	80
5.3.3	Memorian biltegiraketa . . . . .	81
5.3.4	Motze-errorearen azterketa . . . . .	82
5.4	Nodo distantzikideak: Diferentzia finituak . . . . .	83
5.4.1	Errorearen zabaltzea . . . . .	85
5.5	Hermite polinomioa . . . . .	87
5.6	Chebyshev nodoak . . . . .	90
5.7	Runge efektua . . . . .	93
5.8	Funtzio zatika polinomikoen bidezko interpolazioa . . . . .	93
	<b>Bibliografia</b>	<b>94</b>



---

# 1. kapitulua

---

## Errorea eta algoritmoen ezaugarriak

### 1.1 Errorearen jatorriak

- Problemaren planteamenduan
  - problemaren modelizazioa
  - datu fisikoak
- Zenbakizko ebazpenean
  - motze-errorea
  - borobiltze-errorea

#### 1.1.1 Motze-errorea

**1.1 adibidea.**  $f(x) = \sin x$  funtzioaren Taylor garapena zera da:

$$\sin x = \sum_{i=0}^{\infty} (-1)^i \frac{x^{2i+1}}{(2i+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Praktikan, seriea moztu behar dugu batugai kopuru finitu bat batzeko. Demagun  $n$ -garren batugaian gelditzen garela, honako polinomio hau sortuz:

$$p_n(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

$p(x) \neq \sin x$  denez, egin dugun *motze errorea* zera da:

$$e_t = \sin x - p_n(x).$$

**1.2 adibidea.** Demagun  $f: [0, 1] \rightarrow \mathbb{R}$  funtzio integragarri bat dela. Izan bedi

$$I(f) = \int_0^1 f(x) dx$$

$I(f)$  balioaren hurbilpen bat kalkula daiteke *Riemann baturak* erabiliz.  $f$ -ren grafikoak eta ardatzak mugatzen duten azaleraren balioa hurbildu daiteke  $n$  laukizuzen erabiliz,  $Q(f)$  haien azalaren batura izanik:

$$Q(f) = \frac{1}{n} \left( \frac{f(0) + f(1/n)}{2} + \frac{f(1/n) + f(2/n)}{2} + \dots + \frac{f((n-1)/n) + f(1)}{2} \right).$$

Egindako motze-errorea:  $e_t = I(f) - Q(f)$

### 1.1.2 Borobiltze errorea

Ordenagailu bat erabiltzen badugu aurreko adibideko  $Q(f)$  kalkulatzeko, beraz, makina-zenbakiak erabiltzeagatik, hau da, hamartar kopurua mugatuta egonik, ez dugu  $Q(f)$  balioa lortuko, baizik eta  $\widetilde{Q}(f)$  beste balio bat. Egindako *borobiltze-errorea* zera da:

$$e_r = Q(f) - \widetilde{Q}(f).$$

### 1.1.3 Errore totala

Aurreko adibideko integralaren balioari ordenagailuz lortzen dugun balioa kentzen badiogu, errore totala lortzen dugu:

$$\begin{aligned} e &= I(f) - \widetilde{Q}(f) \\ &= I(f) - Q(f) + Q(f) - \widetilde{Q}(f) \\ &= e_t + e_r \end{aligned}$$

### 1.1.4 Hasierako errorea

Demagun  $\sin \sqrt{3}$  kalkulatu nahi dugula  $p_n$  Taylor polinomioa erabiliz, eta kalkulurako,  $\sqrt{3} \approx 1.7$  hartzen dugula. Hasierako errore bat eragiten du hurbilketa honek:

$$e_i = p_n(\sqrt{3}) - p_n(1.7)$$

Kasu honetan, errore guztiak metatzen dira errore totalean:

$$\begin{aligned} e &= \sin \sqrt{3} - \widetilde{p_n(1.7)} \\ &= \sin \sqrt{3} - p_n(\sqrt{3}) + p_n(\sqrt{3}) - p_n(1.7) + p_n(1.7) - \widetilde{p_n(1.7)} \\ &= e_t + e_i + e_r. \end{aligned}$$

## 1.2 Errore absolutu eta erlatiboa

**1.1 definizioa.** Benetazko balio bat eta bere hurbilketa baten arteko diferentziari *errore absolutu* esaten zaio, eta honela idatziko da:

$$e = x - \bar{x},$$

benetazko balioa eta hurbilketa  $x$  eta  $\bar{x}$  izanik, hurrenez hurren.

Balio negatiboak ekiditzeko, ohikoa da balio absolutua hartzea, hau da:

$$e = |x - \bar{x}|.$$

**1.2 definizioa.** Errore absolutua eta benetazko balioaren arteko zatidurari *errore erlatibo* esaten zaio, eta honela idazten da:

$$e_{rel} = \frac{e}{|x|} = \frac{|x - \bar{x}|}{|x|}$$

Askotan,  $|x|$  balioa ezezagutzen ez dugunez, ohikoa da  $|\bar{x}|$  balio ezagunaz zatitzea errore erlatiboa kalkulatzeko, hau da:

$$e_{rel} = \frac{|x - \bar{x}|}{|\bar{x}|}.$$

Errore erlatiboa adieraz daiteke portzentaiezko adierazpena erabiliz:  $\%100 \cdot e_{rel}$ .

**1.3 adibidea.** Demagun  $x = 0.00006$  balio zehatza dela eta  $\bar{x} = 0.00005$  bere hurbilketa. Egindako errore absolutua txikia dirudi:

$$e = |x - \bar{x}| = |0.00006 - 0.00005| = 0.00001 = 10^{-5}$$

baina errore erlatiboa handia da:

$$e_{rel} = \frac{|x - \bar{x}|}{|\bar{x}|} = \frac{10^{-5}}{0.00005} = 0.2 \equiv \%20a.$$

**1.4 adibidea.** Demagun  $x = 100500$  balio zehatza dela eta  $\bar{x} = 100000$  bere hurbilketa. Orain, errore absolutua handia dirudi:

$$e = |x - \bar{x}| = 500$$

baina errore erlatiboa txikia da:

$$e_{rel} = \frac{|x - \bar{x}|}{|\bar{x}|} = \frac{500}{100000} = 0.005 \equiv \%0.5a.$$

## 1.3 Koma flotatzaileko adierazpena

### 1.3.1 Borobiltze-prozesua

Ordenagailu batek bakarrik biltegira dezake informazio mugatua. Zenbakiak idazteko, *koma flotatzaileko adierazpena* erabiliko dugu:

$$x = \pm(0.d_1d_2 \dots d_t) \cdot \beta^e,$$

non honako osagai hauek ditugu:

- $e \in \mathbb{Z}$  berretzailea da;
- $\beta \in \mathbb{N} - \{1\}$  oinarria da;
- $d_i \in \{0, 1, \dots, \beta - 1\}$  digituak dira;
- $0.d_1d_2 \dots d_t \in [0, 1)$  zenbakiari *mantisa* deritzo, zeinen digituak  $\beta$  oinarrian adierazita dauden, hau da,

$$0.d_1d_2 \dots d_n \equiv d_1 \cdot \beta^{-1} + d_2 \cdot \beta^{-2} + \dots + d_t \cdot \beta^{-t}.$$

- $t \in \mathbb{N}$  *mantisaren luzera* da, eta digitu esanguratsu kopurua da.

**1.3 definizioa.** Ordenagailuak erabiltzen dituen zenbakiak *makina-zenbakiak* direla esango dugu, eta honako adierazpen hau erabiliko da:  $x$  benetazko zenbakiari  $fl(x)$  makina-zenbakia dagokio.

Hasiera batean, zenbaki batek koma flotatzaileko adierazpen bat baino gehiago onartu dezake:

$$0.003 = 0.3 \cdot 10^{-2} = 0.03 \cdot 10^{-1} = 0.00003 \cdot 10^1 = \dots$$

Hori ekiditzeko, inposatuko diogu mantisari  $d_1 \neq 0$  izatea. Baldintza hori betetzen duten zenbakiei *normalizatuta* esango zaie.

Mantisaren luzeraren arabera, esango dugu ordenagailu batek *zehaztasun simple* (7 digitu esanguratsu, hau da,  $t = 7$ ) edo *zehaztasun bikoitza* (16 digitu esanguratsu, hau da,  $t = 16$ ) erabiltzen duela.

Simplifikatzeko, beti hartuko dugu  $\beta = 10$  (oinarri hamartarra, hau da,  $d_i \in \{0, 1, \dots, 9\}$ ) baita mantisa normalizatuak ere ( $d_1 \neq 0$ ). Honako notazio hauek erabiltzen dira:

$$2432.91 = 0.243291 \cdot 10^4 = 0.243291E4.$$

Mantisaren luzera ( $t$ ) txikiegia gertatzen denean, 2 aukera ditugu adierazteko:

- mozketa hutsa (hartu mantisa teorikoaren lehenengo  $t$  hamartar);
- ohiko borobilketa, hau da,  $d_{t+1} \geq 5$  bada, orduan,  $10^{-t}$  gehitzen zaio mantisari.

**1.5 adibidea.**  $x = 2/3 = 0.66666\dots$  zenbakia adieraziko dugu  $t = 4$  digitu esanguratsuz goiko bi teknikak aplikatuz: mozketa hutsa eginez,  $fl(x) = 0.6666$  lortuko genuke, eta borobilduz, aldiz,  $fl(x) = 0.6667$ . Mozketaz,  $0.00006666\dots$  errorea egin dugu, eta borobilduz, aldiz,  $0.00003333\dots$

Demagun  $d_{t+1} \leq 4$  dela. Orduan,  $fl(x)$  berdina lortzen da moztuz zein borobilduz. Aldiz,  $d_{t+1} \geq 5$  bada, orduan mozketa egindako errorea, borobilduz egiten dena baino handiagoa da (aurreko adibidean gertatzen denez). Horregatik, beti erabiliko dugu ohiko borobilketa.

**1.1 adibideak.** • Borobiltzen badugu  $\pi = 3.14159265\dots$  zenbakia  $t = 4$  digitu esanguratsutara, orduan  $fl(\pi) = 3.142$  da.  $t = 5$  erabiliz, aldiz,  $fl(\pi) = 3.1416$ .

- $x = 10.18999$  bada eta  $t = 6$  erabiltzen badugu, orduan,  $fl(x) = 10.19$ .
- Borobiltzen badugu  $x = \sqrt{2} = 1.41421\dots$  4 zifra esanguratsutara, 1.414 lortuko dugu. Borobiltzen badugu 4 zifra dezimaletara, orduan 1.4142 lortuko dugu.

### 1.3.2 Ordenagailuaren zehaztasuna

Bedi  $x \in \mathbb{R}$ . Erabiliko dugu  $t$  zifra esanguratsu  $x$  adierazteko, eta  $fl(x)$  makina-zenbakia kontsideratuz egiten den errorea neurtuko dugu. Badaukagu

$$fl(x) = \alpha \cdot 10^e,$$

non  $\alpha$  mantisa normalizatua baita. Badaukagu, ere,

$$x = a \cdot 10^e.$$

Borobiltze prozesuagatik, erraz ikus daiteke  $|a - \alpha| \leq 0.\overbrace{0\dots0}^t 5 = 5 \cdot 10^{-t-1}$  dela. Orduan, borobilduz egiten dugun errore absolutua borna dezakegu honela:

$$e = |x - fl(x)| = |a \cdot 10^e - \alpha \cdot 10^e| = 10^e \cdot |a - \alpha| \leq 5 \cdot 10^{e-t-1}$$

Adierazpen normalizatua erabiltzeagatik, badaukagu  $|a| \geq 0.1$ . Hori erabiliz, borna dezakegu errore erlatiboa:

$$e_{rel} = \frac{|x - fl(x)|}{|x|} \leq \frac{5 \cdot 10^{e-t-1}}{|a \cdot 10^e|} \leq \frac{5 \cdot 10^{e-t-1}}{10^{-1} \cdot 10^e} = \boxed{5 \cdot 10^{-t}}$$

Azken borne honi deituko diogu *ordenagailuaren zehaztasuna*, eta  $eps = 5 \cdot 10^{-t}$  adierazpenaz idatziko dugu.

**1.1 lema.**  $x \in \mathbb{R}$  guztietarako existitzen da  $\varepsilon_x \in \mathbb{R}$  zeinerako

$$fl(x) = x \cdot (1 + \varepsilon_x)$$

dugun,  $|\varepsilon_x| \leq eps$  izanik.

*Frogapena.* Hartu  $\varepsilon_x = \frac{fl(x)-x}{x}$ . ♣

Normalean, ordenagailu batean  $e$  berretzailearen balio posibleak mugatuta daude. Adibidez,  $-99 \leq e \leq 99$  izan daiteke. Kasu horretan,  $x = 0.3421234E110$  ez litzateke makina-zenbakirik izango, eta *overflow* gertatzen dela esango genuke. Bestalde,  $y = 0.234321E-110$  zenbakiari dagokion makina-zenbakia  $fl(y) = 0$  izango litzateke, eta *underflow* gertatzen dela esango genuke.

## 1.4 Koma flotatzaileko aritmetika

Demagun makina batek  $t$  zifra esanguratsuz lanegiten duela. Eragiketa **bakoitza** egin ondoren, borobilketa egiten du, eta honek ondorio garrantzitsuak dakartza azken emaitzerako.



### 1.4.1 Batuketa ez da elkarkorra

Ohiko aritmetikan, batuketa elkarkorra da, hau da,  $a + (b + c) = (a + b) + c$ ,  $\forall a, b, c \in \mathbb{R}$ . Baina koma flotatzaileko aritmetika erabiltzen dugunean, ordena inportantea da, eta beraz,  $fl(fl(a + b) + c) \neq fl(a + fl(b + c))$  izan daiteke.

**1.6 adibidea.** Kontsidera dezagun  $t = 4$  eta honako zenbaki hauek:

$$a = 0.002333, \quad b = 55.55, \quad c = -55.45$$

Badaukagu

$$fl(b + c) = 0.1 \implies fl(a + fl(b + c)) = fl(0.102333) = 0.1023$$

Bestalde,

$$fl(a+b) = fl(55.552333) = 55.55 = b \implies fl(fl(a+b)+c) = fl(b+c) = 0.1 \neq 0.1023$$

### 1.4.2 Elementu neutroa ez da bakarra

Aritmetika arrunta erabiliz,  $x + y = x \implies y = 0$  ondorioztatzen da, baina koma flotatzaileko adierazpenaz, baliteke infinitu  $y$  existitzea zeintzuetarako  $fl(x + y) = x$  baita.

**1.7 adibidea.**  $t = 3$  hartuz,

$$x = 0.234E7, \quad y = 0.123E-2 \neq 0$$

eta badaukagu

$$fl(x + y) = fl(2340000.00123) = 0.234E7 = x.$$

Ondorioz, eragiketak gauzatzeko, garrantzi handia izango du ordenak errorea hedatu ez dadin. Batugaiak ordenatzeko estrategia bereziak ikusiko ditugu:

**1.1 trikimailua.** Urrezko legetzat har dezakegu honako baieztapen informal hau: “Koma flotatzaileko aritmetikan, inoiz ere ez gehitu zenbaki handi bat txikiegia den beste bati”. Aurreko adibideetan ikusi dugunez, zifra esanguratsu galtzen dira prozesu horretan.

**1.2 trikimailua.** Zenbaki batzu batu behar baditugu, hobe dugu txikienetatik hasi, honako adibide honetan frogatuko dugunez:

Izan bitez  $a_1, a_2, a_3, a_4, a_5 \in \mathbb{R}$  eta demagun koma flotatzaileko aritmetika erabiliz batzen ditugula honako ordena hau jarraituz:

$$((((a_1 + a_2) + a_3) + a_4) + a_5)$$

Orduan, koma flotatzaileko adierazpena erabiliz, benetazko emaitza honako hau izango litzateke:

$$\begin{aligned} \text{emaitza} &= fl(fl(fl(fl(a_1 + a_2) + a_3) + a_4) + a_5) \\ &= (((a_1 + a_2)(1 + \varepsilon_1) + a_3)(1 + \varepsilon_2) + a_4)(1 + \varepsilon_3) + a_5)(1 + \varepsilon_4) \end{aligned}$$

Simplifikatzeko, suposatuko dugu  $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon$  direla, non  $|\varepsilon| \leq \text{eps}$ . Biderketak burutuz, badaukagu:

$$\begin{aligned} \text{emaitza} &\simeq a_1(1 + \varepsilon)^4 + a_2(1 + \varepsilon)^4 + a_3(1 + \varepsilon)^3 + a_4(1 + \varepsilon)^2 + a_5(1 + \varepsilon) \\ &= (a_1 + a_2 + a_3 + a_4 + a_5) + \varepsilon(4a_1 + 4a_2 + 3a_3 + 2a_4 + a_5) + \overbrace{\varepsilon^2(\dots)}^{O(\varepsilon^2)} + \dots, \end{aligned}$$

non Landau O larria notazioa erabili dugun (hondarra zerora  $\varepsilon^2$  bezain agudo doala adierazteko). Hondar hori meziprezugarritzat eman dezakegu. Laburbilduz, frogatu dugu

$$\begin{aligned} \text{emaitza} &\simeq (a_1 + a_2 + a_3 + a_4 + a_5) + \varepsilon(4a_1 + 4a_2 + 3a_3 + 2a_4 + a_5) \\ &\implies \text{errorea} = \varepsilon(4a_1 + 4a_2 + 3a_3 + 2a_4 + a_5). \end{aligned}$$

Era berean froga daiteke  $n$  batugai erabiltzen dugunean, orduan

$$\begin{aligned} \text{emaitza} &\simeq (a_1 + \dots + a_n) + \varepsilon((n-1)a_1 + (n-1)a_2 + (n-2)a_3 + \dots + 2a_{n-1} + a_n) \\ &\implies \text{errorea} = \varepsilon((n-1)a_1 + (n-1)a_2 + (n-2)a_3 + \dots + 2a_{n-1} + a_n). \end{aligned}$$

Nabaria denez, lehenengo bi gaiek eragindako errorea hirugarrenak egindakoa baino handiagoa dela, eta eragin horrek gero eta baxuagoa da. Horregatik hobe dugu batugaiak ordena gorakorrez antolatu batu baino lehen.

**1.8 adibidea.**  $\sum_{n=1}^{500} \frac{1}{n^5}$  batzeko ordenarik egokiena honako hau da:

$$\frac{1}{500^5} + \frac{1}{499^5} + \frac{1}{498^5} + \dots$$

### 1.4.3 Deuseztapen efektua

Demagun  $x$  eta  $y$  oso antzekoak direla eta  $x - y$  kalkulatu nahi dugula koma flotatzaileko aritmetikaz. Orduan, dijitu esanguratsu asko gal ditzakegu, errore erlatibo handia gertatuz, honako adibide honetan ikus dezakegunez:

**1.9 adibidea.** Bitez  $x = 0.123444444$  eta  $y = 0.123411111$ . Orduan, badaukagu  $x - y = 0.33333E-4$ . Bestalde,  $t = 5$  digitu esanguratsuz lanegiten duen makina batek ez luke lortuko  $x - y$  balio zehatza, baizik eta:

$$\alpha = fl(fl(x) - fl(y)) = fl(0.12344 - 0.12341) = fl(0.00003) = 0.3000E-4$$

Nabaritu  $\alpha$ -ren mantisako lehen digitua bakarrik dela zehatza. Beraz,  $x - y$  kalkulatzeko egindako errore erlatiboa  $x$  borobiltzean egindakoa baino askoz handiagoa da.

$$e_{relx} = \frac{|fl(x) - x|}{|x|} = 0.000036 \equiv \%0.0036, \quad e_{rel(x-y)} = \frac{|\alpha - (x - y)|}{|x - y|} = 0.09999 \dots \approx \%10$$

Orain ikusiko dugu deuseztapen efektua ekiditzeko hainbat teknika:

**1.3 trikimailua.** Batzuetan, **konjokatu egokiaz biderkatzea eta zatitzea** komenigarri gertatzen da, honako adibide honetan ikus dezakegunez: Demagun balioztatu nahi dugula

$$f(x) = 1 - \cos x$$

funtzioa, 0-ren hurbileko  $x$  batean. Orduan,  $x \sim 0 \implies \cos x \sim 1$ , eta deuseztapen efektuaren arriskuan egongo gara. Hori ekiditzeko, erabil dezakegu beste adierazpen baliokide bat, zein deuseztapen efektuaren arriskupean ez baitago:

$$1 - \cos x = (1 - \cos x) \cdot \frac{1 + \cos x}{1 + \cos x} = \frac{1 - \cos^2 x}{1 + \cos x} = \boxed{\frac{\sin^2 x}{1 + \cos x}}$$

Adibidez,  $t = 5$  hartzen badugu, eta  $x = 0.14500E-1$  kontsideratuz, badaukagu

$$\cos x = 0.99989E0 \implies 1 - \cos x = 0.11000E-3,$$

bakarrik lehen digitua zehatza izanik. Bestalde,

$$1 + \cos x = 0.19999E1, \quad \sin x = 0.14499E-1 \implies \sin^2 x = 0.21022E-3$$

$$\implies \frac{\sin^2 x}{1 + \cos x} = fl\left(\frac{0.00021022}{1.9999}\right) = 0.10512E-3,$$

zeinen hamartar guztiak zehatzak baitira, Taylor seriez konproba dezakegunez:

$$f(x) = 1 - \cos x = \frac{x^2}{2} - \frac{x^4}{4!} + \frac{x^6}{6!} - \dots$$

**1.4 trikimailua.** Demagun 2. mailako ekuazioa ebatzi nahi dugula:

$$ax^2 + bx + c = a(x - x_1)(x - x_2) = 0 \implies x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (1.1)$$

Demagun  $b^2 \gg 4|ac|$  dela. Orduan, badaukagu  $\sqrt{b^2 - 4ac} \approx |b|$ , eta erro bat kalkulatzeko formula deuseztapen efektuaren arriskuan egon daiteke. Hori ekiditzeko, badaukagu 2 estrategia. Sinplifikatzeko, suposatuko dugu  $b > 0$  dela.

i)

$$\begin{aligned} x_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \cdot \frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}} \\ &= \frac{b^2 - b^2 + 4ac}{2a(-b - \sqrt{b^2 - 4ac})} = \boxed{\frac{-2c}{b + \sqrt{b^2 - 4ac}}}, \end{aligned}$$

eta azken adierazpen horrek ez dauka deuseztapen efekturik, ezta  $x_2$  kalkulatzeko ohiko formulak ere.

ii) Beste estrategia bat izan daiteke honako hau: kalkula dezakegu  $x_2$  (ez dago deuseztapen efekturik):

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}.$$

Gero, (1.1) formulatik, badaukagu

$$x_1 \cdot x_2 = \frac{c}{a} \implies \boxed{x_1 = \frac{c}{x_2 \cdot a}}.$$

Nabaritu  $x_1 = -x_2 - b/a$  formula desegokia dela deuseztapen efektuagatik.

## 1.5 Algoritmoen propietateak

Algoritmo bat zehazki definituriko prozedura finitu bat da, normalean etapatan antolatuta dago eta erraz idazten da adierazpen formala erabiliz. Horregatik, oso aproposak dira makina batekin programatzeko.

### 1.5.1 Konbergentzia

Algoritmo askotan,  $\{u_n\}_{n \in \mathbb{N}}$  segida bat sortzen da,  $u_n$  gaia algoritmoko  $n$ -garren pausuan lortzen den emaitza izanik. Baldintza batzuk betetzen badira, segida horrek problemaren soluzio zehatzera konbergitzen du. Konbergentziaren *abiadura* oso garrantzitsua izango da, ere.

## 1.5.2 Egonkortasuna eta problemaren baldintzatzea

Nahiz eta algoritmo bat konbergente eta azkarra izan, horrek ez du ezertarako balio borobiltze-errorea kontrolatuta ez badago.

Algoritmo baten zehar borobiltze-erroreak gehiegi hedatzen badira, irteerako emaitzak nahiko okerrak izan daitezke. Kasu horretan, esaten da algoritmoa *desegonkorra* dela. Bestela, borobiltze-errorea kontrolatzen bada, esango dugu algoritmoa *egonkorra* dela.

Bestalde, baliteke algoritmo bat oso “sentibera” izatea honako sentzu honetan: hasierako baldintzetako aldaketa txikiek eragin handia badaukate bukaerako emaitzetan, esango dugu algoritmoa edo problema *txarto baldintzatuta* dagoela.

### 1.10 adibidea.

$$\begin{cases} 2x & +6y & =8 \\ 2x & +6.00001y & =8.00001 \end{cases} \text{ sistemaren ebazpen zehatza } \begin{cases} x = 1 \\ y = 1 \end{cases} \text{ da.}$$

$$\begin{cases} 2x & +6y & =8 \\ 2x & +5.99999y & =8.00002 \end{cases} \text{ sistemaren ebazpen zehatza } \begin{cases} x = 10 \\ y = -2 \end{cases} \text{ da.}$$

Hastapen baldintzen bariazioa  $10^{-5}$  ordenakoa izan da, baina emaitzaren bariazioa, aldiz,  $10^1$  ordenakoa. Sistema hori *txarto baldintzatuta* dago.

### 1.11 adibidea (Wilkinson, 1959).

$$\begin{aligned} p(x) &= \prod_{i=1}^{20} (x - i) = (x - 1) \cdot (x - 2) \dots (x - 20) \\ &= x^{20} - 210x^{19} + 2065x^{18} - \dots + 20! \\ &= a_0x^{20} + a_1x^{19} + \dots + a_{20}. \end{aligned}$$

polinomioaren erroak  $1, 2, \dots, 20$  dira. Sortuko dugu  $q(x)$  polinomio berri bat  $p(x)$ -ren koefiziente bat pixkat aldatuz honela:

$$q(x) = a_0x^{20} - (210 + 10^{-7})x^{19} + a_2x^{18} + \dots + a_{20}.$$

Polinomio berri horren erroak honako hauek dira:

$$1, 2, 3, 4, 4.999999939, 6.000005825, 6.999745979, \dots \\ \dots, 16.72503985 \pm 2.731576697i, 19.45963983 \pm 1.87435706i, 20.788880562.$$

Esate baterako, polinomio baten erroak kalkulatzeko bere koefizienteekiko txarto baldintzatua dago. Ondorioz, zenbakizko ikuspuntutik matrize baten autobalioak polinomio karakteristikoaren bidez kalkulatzeko metodoa **txarto baldintzatua** dago, eta beraz, beste metodoak erabiliko ditugu.

### 1.5.3 Eragiketa-kostua

Komenigarria da ahalik eta eragiketa gutxien egitea, ekonomia, denbora eta zehaztasunagatik. Algoritmo bakoitzean aztertuko dugu, ba, eragiketa-kostua. Orokorrean, eragiketa kostuak kalkulatzeko, batuketak ez dira kontuan hartzen, biderketa batekin konparatuz batuketa bat egiteko denbora mesprezugarritzat hartzen baita.

#### Hörner algoritmoa

Metodo bikaina da polinomio bat eta bere deribatuak puntu jakin batean balioztatzeko, eragiketa kostua ikuspuntutik. Honako berridazketa honetan dago oinarrituta:

$$\begin{aligned} p(x) &= a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n \\ &= ((\dots((a_0x + a_1)x + a_2)x + \dots)x + a_{n-1})x + a_n \end{aligned}$$

Ohiko adierazpenaz,  $p(x)$  ebaluatzeko,  $n + (n-1) + \dots + 2 + 1 \simeq n^2/2$  biderketa eta  $n$  batuketa egin behar dira. Bestalde, bigarren adierazpenaz, bakarrik  $n$  biderketa eta  $n$  batuketa dira beharrezkoak.

Hörner algoritmoaren deskribapena zera da: Demagun  $p(x)$  polinomioa  $r$  puntuan ebaluatu nahi dugula. Orduan,  $a_0, \dots, a_n$  koefizienteekin honako segida hau sor daiteke:

$$\begin{aligned} b_0 &= a_0 \\ b_1 &= b_0 \cdot r + a_1 \\ b_2 &= b_1 \cdot r + a_2 \\ &\dots \\ b_k &= b_{k-1} \cdot r + a_k \\ &\dots \\ b_n &= b_{n-1} \cdot r + a_n \end{aligned}$$

Ondorioz,  $p(r) = b_n$ . Algoritmoa deskriba daiteke honako era trinko honetan:

$$\begin{cases} b_0 = a_0 \\ b_i = b_{i-1} \cdot r + a_i, & i = 1, 2, \dots, n. \end{cases}$$

Sorturiko  $b_0, \dots, b_{n-1}$  koefizienteak erabiliko ditugu  $p'(r)$  kalkulatzeko. Horretarako, honako eragiketa aljebraiko honetaz baliatuko gara:  $p(x)$  polinomioa  $x - r$  polinomiotik zatituz,  $q_{n-1}(x)$  zatidura eta  $R_0 \in \mathbb{R}$  hondarra lortzen dira, hau da

$$p(x) = (x - r) \cdot q_{n-1}(x) + R_0.$$

**1.2 lema.** *Aurreko baldintzetan,  $R_0 = b_n$  eta*

$$q_{n-1}(x) = b_0x^{n-1} + b_1x^{n-2} + \dots + b_{n-1} \quad (1.2)$$

*Frogapena.*

$$\begin{aligned} & (x-r) \cdot (b_0x^{n-1} + b_1x^{n-2} + \dots + b_{n-1}) + b_n \\ &= b_0x^n + (b_1 - r \cdot b_0)x^{n-1} + (b_2 - r \cdot b_1)x^{n-2} + \dots + (b_{n-1} - r \cdot b_{n-2})x + (b_n - r \cdot b_{n-1}) \\ &= a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n \\ &= p(x) \end{aligned}$$



Orain, (1.2) deribatuz,

$$p'(x) = (x-r) \cdot q'_{n-1}(x) + q_{n-1}(x) \implies p'(r) = q_{n-1}(r),$$

eta erabiliko dugu Hörner metodoa  $q_{n-1}(r)$  kalkulatzeko:

$$\begin{aligned} c_0 &= b_0 \\ c_1 &= c_0 \cdot r + b_1 \\ c_2 &= c_0 \cdot r + b_2 \\ &\dots \\ c_k &= c_{k-1} \cdot r + b_k \\ &\dots \\ c_{n-1} &= c_{n-2} \cdot r + b_{n-1} = q_{n-1}(r) = p'(r) \end{aligned}$$

Are gehiago,  $p''(r)$  kalkulatzeko, aurreko lema eta emaitzak erabil ditzakegu. Hasteko, zatitu dezakegu  $q_{n-1}(x)$  polinomioa  $(x-r)$ -tik, honako hau lortuz:

$$q_{n-1}(x) = (x-r) \cdot q_{n-2}(x) + R_1, \quad (1.3)$$

non  $R_1 \in \mathbb{R}$  eta  $q_{n-2}(x) = c_0x^{n-2} + c_1x^{n-3} + \dots + c_{n-3}x + c_{n-2}$ . Orduan, (1.2) eta (1.3) erabiliz,

$$\begin{aligned} p(x) &= (x-r)^2 q_{n-2}(x) + R_1(x-r) + R_0 \\ p'(x) &= 2(x-r)q_{n-2}(x) + (x-r)^2 q'_{n-2}(x) + R_1 \\ p''(x) &= 2q_{n-2}(x) + 4(x-r)q'_{n-2}(x) + (x-r)^2 q''_{n-2}(x) \\ \implies p''(r) &= 2 \cdot q_{n-2}(r). \end{aligned}$$

Jakina,  $q_{n-2}(r)$  kalkulatzeko, Hörner algoritmoa erabiliko genuke,  $c_i$  koefizienteak erabiliz. Orokorrean, goiko ordenako deribatuak ebaluatzeko, prozesua  $k$  aldiz errepikatuz, badaukagu

$$p^{(k)}(r) = k!q_{n-k}(r)$$

Komenigarria da kalkuluak antolatzea Ruffini ebaluaketa moduan.

ADIBIDEA

### 1.5.4 Memoriaren erabilera

Komenigarria da ahalik eta memoria gutxien erabiltzea algoritmo bat burutzeko. Egoera optimoan, INPUT gordetzeko memoria espazioa besterik ez da erabiltzen OUTPUT lortu eta biltegitratzeko. Horretarako, urrezko lege orokor bat dago: prozesu baten zehar, datu bat beharrezkoa baldin ez bada, orduan ezaba dezakegu eta hutsik uzten duen lekua aprobeitza dezakegu behar izango dugun beste datu bat biltegitratzeko. Adibidez, ikusirik Höner algoritmoaz kalkula eta biltegitra daitezke  $p(r), p'(r), \dots, p^n(r)$  balioak  $a_0, \dots, a_n$  koefizienteak biltegitratzeko behar dugun  $n + 1$  luzerako bektore bat erabiliz. (ADIBIDEA)

### 1.5.5 Metodo motak

- Metodo zuzenak
  - Teoriaz, ebazpen zehatza lortzen da.
  - Borobilduz errorea heda daiteke.
- Iteraziozko metodoak
  - Hurbilpenak lortzen dira etapa bakoitzean  $\{u_n\}_{n \in \mathbb{N}} \rightarrow u$
  - borobiltze-errorea ez da hain inportante. Konbergentzia nahitaezkoa da.



---

## 2. kapitulua

---

# Ekuaizio linealezko sistemen ebazpena

### 2.1 Sarrera

Erabiliko dugun notazioaren arabera,  $\mathbb{R}^n$  espazioko bektoreak zutabe gisan idatziko ditugu:

$$\underline{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

Errenkada gisan idazteko, irauli behar dugu:

$$\underline{x}^t = (x_1, x_2, \dots, x_n).$$

Badaukagu  $\mathbb{R}^n$  espazioko oinarri kanonikoa:

$$\beta = \{\underline{e}_1, \underline{e}_2, \dots, \underline{e}_n\}, \quad \underline{e}_i = (0, \dots, \overset{i}{\hat{1}}, \dots, 0)^t \text{ izanik.}$$

**2.1 definizioa.** Esango dugu  $U = (u_{ij}) \in \text{Mat}(n \times n, \mathbb{R})$  matrizea *goi-triangeluarra* dela baldin eta  $u_{ij} = 0$  bada  $\forall i > j$ .

$$U = \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & u_{nn} \end{pmatrix}$$

**2.2 definizioa.** Esango dugu  $L = (l_{ij}) \in \text{Mat}(n \times n, \mathbb{R})$  matrizea *behe-triangeluarra* dela baldin eta  $u_{ij} = 0$  bada  $\forall i < j$ .

$$L = \begin{pmatrix} l_{11} & 0 & \dots & \dots & 0 \\ l_{21} & l_{22} & \ddots & & 0 \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \dots & \dots & l_{nn} \end{pmatrix}$$

**2.3 definizioa.** Baldin  $D \in \text{Mat}(n \times n, \mathbb{R})$  matrizea goi-triangeluar eta behe-triangeluarra bada, orduan  $D$  matrize *diagonal* bat dela esango dugu.

$$D = \begin{pmatrix} d_{11} & 0 & \dots & 0 \\ 0 & d_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_{nn} \end{pmatrix}$$

**2.1 teorema.** Bedi  $A = (a_{ij}) \in \text{Mat}(n \times n, \mathbb{R})$  matrize *triangeluar* bat (goi-, behe-edo *diagonal*). Orduan,

$$|A| = \det(A) = \prod_{i=1}^n a_{ii} = a_{11} \cdot a_{22} \cdot \dots \cdot a_{nn}$$

**2.2 teorema.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$ . Orduan, honako propietate hauek baliokideak dira:

- $A$ -ren alderantzizkoa existitzen da ( $\exists A^{-1}$ );
- $A$  ez-singular da, i.e.  $|A| \neq 0$ ;
- $A \cdot \underline{x} = \underline{0} \implies \underline{x} = \underline{0}$ ;
- $A \cdot \underline{x} = \underline{b}$  sistemak soluzio bakarra du  $\forall \underline{b} \in \mathbb{R}^n$
- $A$  matrizea osatzen duten errenkada-bektoreak linealki independenteak dira.
- $A$  matrizea osatzen duten zutabe-bektoreak linealki independenteak dira.

**Problema orokorra:**

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Goiko sistema era matrizialean adieraz dezakegu:

$$A \cdot \underline{x} = \underline{b},$$

non

$$A = (a_{ij}), \quad \underline{x}^t = (x_1, \dots, x_n), \quad \underline{b}^t = (b_1, \dots, b_n)$$

diren.

Nabaritu  $A$  matrizearen alderantzizkoa aurkitzea eta  $A \cdot \underline{x} = \underline{b}$  motako sistemak ebatzea problema baliokideak direla honako zentzu honetan:  $A$  matrizearen alderantzizkoa kalkulatzeko badakigu, orduan  $A \cdot \underline{x} = \underline{b}$  sistema ebatz dezakegu honako era honetara:

$$A \cdot \underline{x} = \underline{b} \Leftrightarrow A^{-1} \cdot A \cdot \underline{x} = A^{-1} \cdot \underline{b} \Leftrightarrow \underline{x} = A^{-1} \cdot \underline{b}.$$

Bestalde,  $A^{-1}$  kalkula dezakegu  $n$  sistema ebatziz: hartu aintzakotzat honako matrize hau

$$X = (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n),$$

eta idatz dezagun honako sistema matrizial hau:

$$A \cdot (\underline{x}_1, \underline{x}_2, \dots, \underline{x}_n) = A \cdot X = I_n = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & & & 0 \\ 0 & \dots & 0 & 1 \end{pmatrix} = (\underline{e}_1, \underline{e}_2, \dots, \underline{e}_n)$$

Orduan, badauzkagu  $n$  sistema planteatuta, hau da:

$$A \cdot \underline{x}_i = \underline{e}_i, \quad i = 1, 2, \dots, n.$$

Goiko sistema horiek ebatziz, lortuko genuke  $X = A^{-1}$  matrizea. Sarrera hau amaitzeko, ikus ditzagun praktikan agertzen diren bi matrize motak:

**Neurrizko ordenako matrize dentsoak** Matrize horien tamaina  $n < 100$  edo  $n < 200$  ordenakoa da. Gai gehienak eznuluak izanik eta matrize handiegia ez izanik, matrize osoa biltegitzen da memorian, eta metodo *zuzenak* erabiltzen dira (kontrolatu behar da borobiltze-errorea).

**Ordena handiko matrize sakabanatuak** Tamainu handikoak ( $n = 1000$ , edo handiagoa!). Gai eznulu gutxi daude (sakabanatuak), eta orokorrean, diagonal nagusiaren inguruan metatzen dira. Horregatik, ez dago matrize osoa memoriari biltegitzeko beharrik. Matrize hauekin lanegiteko, *iteraziozko* metodoak erabiltzen dira (borobiltze errorea ez da hain inportante, baina konbergentzia beharrezkoa da).

## 2.2 Atzerakako ordezkapena

Bedi  $U \in \text{Mat}(n \times n, \mathbb{R})$  goi-triangeluarra, eta hartu aintzakotzat  $U \cdot \underline{x} = \underline{b}$  sistema:

$$\left\{ \begin{array}{l} u_{11} \cdot x_1 + u_{12} \cdot x_2 + \dots + u_{1,n-1} \cdot x_{n-1} + u_{1,n} \cdot x_n = b_1 \\ \phantom{u_{11} \cdot x_1} + u_{22} \cdot x_2 + \dots + u_{2,n-1} \cdot x_{n-1} + u_{2,n} \cdot x_n = b_2 \\ \phantom{u_{11} \cdot x_1} \phantom{+ u_{22} \cdot x_2} \ddots \phantom{+ u_{2,n-1} \cdot x_{n-1}} \phantom{+ u_{2,n} \cdot x_n} \phantom{=} \vdots \\ \phantom{u_{11} \cdot x_1} \phantom{+ u_{22} \cdot x_2} \phantom{+ u_{2,n-1} \cdot x_{n-1}} \phantom{+ u_{2,n} \cdot x_n} \phantom{=} \vdots \\ \phantom{u_{11} \cdot x_1} \phantom{+ u_{22} \cdot x_2} \phantom{+ u_{2,n-1} \cdot x_{n-1}} \phantom{+ u_{2,n} \cdot x_n} \phantom{=} \vdots \\ \phantom{u_{11} \cdot x_1} \phantom{+ u_{22} \cdot x_2} \phantom{+ u_{2,n-1} \cdot x_{n-1}} \phantom{+ u_{2,n} \cdot x_n} u_{n-1,n-1} \cdot x_{n-1} + u_{n-1,n} \cdot x_n = b_{n-1} \\ \phantom{u_{11} \cdot x_1} \phantom{+ u_{22} \cdot x_2} \phantom{+ u_{2,n-1} \cdot x_{n-1}} \phantom{+ u_{2,n} \cdot x_n} \phantom{=} u_{n,n} \cdot x_n = b_n \end{array} \right.$$

Sistema hori ebazteko, *atzerakako ordezkapena* algoritmoa erabiliko dugu, honela deskribatzen dena:

$$\left\{ \begin{array}{l} x_n = \frac{b_n}{u_{n,n}} \\ x_k = \frac{\left( b_k - \sum_{j=k+1}^n u_{k,j} \cdot x_j \right)}{u_{k,k}} \quad k = n-1, n-2, \dots, 1 \end{array} \right.$$

(ADIBIDEA)

### Eragiketa-kostua:

Hemendik aurrera, batuketak eta kendurak egiteko denbora mesprezugarritzat hartuko dugu. Ondorioz, eragiketa kostuak estimatzeko, bakarrik hartuko ditugu kontuan biderketak eta zatiketak.

- $x_n$  kalkulatzeko, zatiketa bat egin behar dugu;

- $x_{n-1}$  lortzeko, biderketa bat eta zatiketa bat;
- $x_{n-2}$  lortzeko, biderketa bi eta zatiketa bat;
- ...
- $x_k$  lortzeko,  $(n - k)$  biderketa eta zatiketa bat;

Guztira,

$$N = 1 + \sum_{k=1}^{n-1} (n - k + 1) = 1 + 2 + \dots + n = \frac{n(n+1)}{2} \simeq \frac{n^2}{2}.$$

## 2.3 Ezabapen Gaussiarra

**Estrategia orokorra:**  $A \cdot \underline{x} = \underline{b}$  sistema ebatzi beharrean, aurkituko dugu  $U \cdot \underline{x} = \underline{c}$  sistema baliokide bat,  $U$  matrizea goi-trianguluarra izanik. Sistema berri hori ebatziko da atzerakako ordezkapena metodoaz.

**Notazioa:** Sistemaren matrize hedatua  $(A|\underline{b})$  notazioaz idatziko dugu.

### 2.3.1 Pibotaketarik gabeko Gauss metodoa

Algoritmo honetako urrats bakoitzean, zutabe batean,  $A$ -ren diagonal nagusiaren azpian dauden osagai guztiak anulatzen dira.  $(A|\underline{b})$  sistema aldatuko da da etapaz etapa,  $n$  etapatan:

$$(A|\underline{b}) = (A^{(1)}|\underline{b}^{(1)}), (A^{(2)}|\underline{b}^{(2)}), \dots, (A^{(n)}|\underline{b}^{(n)}) = (U|\underline{c}),$$

$k$ -garren zutabearen, diagonalaren azpiko osagaiak deuseztatu aurretik daukagun sistema  $(A^{(k)}|\underline{b}^{(k)})$  izanik. Esplizituki,

$$(A^{(k)}|\underline{b}^{(k)}) = \left( \begin{array}{ccccccc|c} a_{11}^{(k)} & a_{12}^{(k)} & \dots & a_{1,k-1}^{(k)} & a_{1,k}^{(k)} & \dots & a_{1,n}^{(k)} & b_1^{(k)} \\ 0 & a_{22}^{(k)} & \dots & a_{2,k-1}^{(k)} & a_{2,k}^{(k)} & \dots & a_{2,n}^{(k)} & b_2^{(k)} \\ 0 & 0 & \ddots & \vdots & \vdots & & \vdots & \vdots \\ \vdots & & \ddots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \dots & a_{k-1,n}^{(k)} & b_{k-1}^{(k)} \\ \vdots & & & 0 & a_{k,k}^{(k)} & \dots & a_{k,n}^{(k)} & b_k^{(k)} \\ \vdots & & & \vdots & a_{k+1,k}^{(k)} & \dots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\ \vdots & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n,k}^{(k)} & \dots & a_{n,n}^{(k)} & b_n^{(k)} \end{array} \right)$$

$(A^{(k)}|\underline{b}^{(k)})$ -tik  $(A^{(k+1)}|\underline{b}^{(k+1)})$  kalkulatzeko, errenkadaz errenkada egingo dira aldatetarik. Bedi  $E_i^{(k)}$ ,  $(A^{(k)}|\underline{b}^{(k)})$  matrize hedatuaren  $i$ -garren errenkada. Esplizituki,

$$E_i^{(k)} = (a_{i,1}^{(k)}, \dots, a_{i,n}^{(k)}|b_i^{(k)}), \quad i = 1, \dots, n.$$

Orduan,  $(k+1)$ -garren matrize hedatuaren errenkadak honako hauek izango dira:

$$E_i^{(k+1)} = \begin{cases} E_i^{(k)} & i = 1, \dots, k \\ E_i^{(k)} - m_{ik} \cdot E_k^{(k)} & i = k+1, \dots, n, \end{cases}$$

non  $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$  zenbakiei biderkatzaileak esango zaien.  $m_{ik}$  biderkatzaileak apropos definituta daude  $(i, k)$  posizioan dagoen osagai deuseztatzeko. Beraz,

$$a_{ik}^{(k+1)} = a_{ik}^{(k)} - m_{ik} \cdot a_{kk}^{(k)} = a_{ik}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \cdot a_{kk}^{(k)} = 0, \quad i = k+1, \dots, n.$$

Algoritmoa deskribatzeko beste era:

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} & \begin{cases} i = 1, \dots, n \\ \forall j \end{cases} \\ 0 & \begin{cases} i = k+1, \dots, n \\ j = 1, \dots, k \end{cases} \\ a_{ij}^{(k)} - m_{ik} \cdot a_{kj}^{(k)} & \begin{cases} i = k+1, \dots, n \\ j = k+1, \dots, n \end{cases} \end{cases}, \quad b_i^{(k+1)} = \begin{cases} b_i^{(k)} & i = 1, \dots, k \\ b_i^{(k)} - m_{ik} \cdot b_k^{(k)} & i = k+1, \dots, n. \end{cases}$$

### Oharrak:

- $a_{kk}^{(k)}$  osagaiari *pibote* esaten zaio. Nulua baldin bada, prozesua gelditu egin behar dugu.
- $m_{ik}$  biderkatzaileak bakarrik behin kalkulatu dira (horregatik idazten dira goi-indizek gabe!), eta gordetzen dira memorian geroago erabiltzeko.
- Erraz frogatu daiteke aljebra lineal erabiliz  $(A^{(k)}|\underline{b}^{(k)})$  sistema hedatu guztiak baliokideak direla (are gehiago,  $|U| = |A^{(k)}| = |A|$ ,  $k = 1, \dots, n$ .)
- $k$ -garren zutabean, diagonalaren azpiko osagaiak deuseztatu ondoren,  $k+1$  lehenengo ekuazioak finkaturik geratzen dira.

**2.3 teorema.** *Suposa dezagun ezabapen Gaussiarra burutu daitekeela  $A \cdot \underline{x} = \underline{b}$  sistema ebazteko (i.e., pibote guztiak eznuiluak direla urrats guztietan). Orduan, badaukagu:*

$$i) |A| = a_{11}^{(1)} \cdot a_{22}^{(2)} \cdot \dots \cdot a_{nn}^{(n)}$$

ii)  $A^{(n)} = U$  idatziz, orduan,  $A = L \cdot U$  faktORIZAZIOA DUGU, non  $L = (l_{ij})$  matrize behe-triangeluar bat den, eta

$$l_{ij} = \begin{cases} 0 & i < j \\ 1 & i = j \\ m_{ij} & i > j \end{cases} \quad \text{den, hau da:} \quad L = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & 0 \\ m_{21} & 1 & 0 & & & \vdots \\ m_{31} & m_{32} & 1 & 0 & & \vdots \\ \vdots & \vdots & & \ddots & \ddots & \vdots \\ \vdots & \vdots & & & 1 & 0 \\ m_{n1} & m_{n2} & \dots & \dots & m_{n,n-1} & 1 \end{pmatrix}.$$

[ADIBIDEAK]

### Memorian gordeketa

Sistema bat ebazteko problemaren INPUT informazioa (hau da, sistemaren matrize hedatua),  $(n \times n + 1)$  matrize bat erabiltzen da. Ikusiko dugun legez, memoriako espazio hori nahikoa izango da sistemaren ebazpena kalkula eta gordetzeko, baita lorturiko  $LU$  faktORIZAZIOA, ere. Gordetzeko prozesua zera da:

1. Hasten gara  $(A|\underline{b})$  matrize hedatuaz;
2.  $k$ -garren urratsean,  $i > k$  bada,  $a_{ik}^{(k)}$  osagaia automatikoki deuseztatzen denez, orduan, horren gainean  $m_{ik}$  biderkatzailea gordeko dugu.
3.  $b_i^{(k+1)}$  idazten da  $b_i^{(k)}$ -ren gainean.
4.  $a_{ij}^{(k+1)}$  osagaia idazten da  $a_{ij}^{(k)}$ -ren gainean  $i = k + 1, \dots, n$  eta  $j = k + 1, \dots, n$  bada.

Azkenean, matrizean izango dugu  $U$  eta  $L$  matrizeak osatzeko informazio guztia. Gero, atzerakako ordezkapena egiterakoan,  $b_i^n$  osagaiaren gainean,  $x_i$  soluzioaren koordinatua gordeko dugu, behetik gora, azken zutabean. Prozesua honako eskema honetan ikus daiteke:

$$\left( A \left| \underline{b} \right. \right) \longrightarrow \left( \begin{array}{ccc} & U & \\ & \vdots & \\ m_{ij} & & \underline{c} \end{array} \right) \longrightarrow \left( \begin{array}{ccc} & U & \\ & \vdots & \\ m_{ij} & & \underline{x} \end{array} \right)$$

**Eragiketa-kostua****Sistema bat ebazteko**

$A \cdot \underline{x} = \underline{b}$  sistema ebazteko prozesua hiru etapatan zatituko dugu:

- $A$  matrizea  $U$  matrize goi-triangeluarra bihurtzeko eragiketa-kostua:  $N_a$ ;
- $\underline{b}$  bektore independentea aldatzeko eragiketa-kostua:  $N_b$ ;
- atzerakako ordezkapenaren eragiketa-kostua:  $N_c$ .

$\boxed{N_a}$  Lehen, zenbatuko ditugu  $k$ -garren etapanegiten diren eragiketak:

- $k$ -garren zutabearen,  $m_{ik}$  biderkatzaile bakoitza kalkulatzeko ( $i = k + 1, \dots, n$ ) zatiketa bat egin behar dugu. Guztira,  $n - k$  eragiketa.
- Biderkatzaileak kalkulatuta,  $a_{ij}^{(k+1)}$  bakoitza kalkulatzeko, biderketa bat egiten da, eta hori egiten da  $(n - k) \times (n - k)$  dimentsioko azpimatrizeko gai bakoitzerako; guztira,  $(n - k)^2$  eragiketa.

Ikusi dugun legez,  $k$ -garren pausuan,  $(n - k) + (n - k)^2$  eragiketa egiten dira. Guzti hau egin behar dugu  $k = 1, 2, \dots, n - 1$  etapetan. Guztira,

$$\begin{aligned} N_a &= \sum_{k=1}^{n-1} [(n - k) + (n - k)^2] = \sum_{i=n-1}^1 (i + i^2) = \sum_{i=1}^{n-1} i + \sum_{i=1}^{n-1} i^2 \\ &= \frac{(1 + n - 1) \cdot (n - 1)}{2} + \frac{(n - 1)n(2n - 1)}{6} \\ &= \frac{3n^2 - 3n + (n^2 - n)(2n - 1)}{6} \\ &= \frac{2n^3 - 2n}{6} \simeq \frac{n^3}{3}, \end{aligned}$$

non  $\sum_{i=1}^n i^2 = n(n + 1)(2n + 1)/6$  formula erabili den.

$\boxed{N_b}$   $b_i^{(k+1)}$  kalkulatzeko, biderkaketa bat egiten da  $i = k + 1, \dots, n$  denean, hau da,  $n - k$  biderkaketa. Etapa guztiak kontuan hartuta,

$$N_b = (n - 1) + (n - 2) + \dots + 2 + 1 = 1 + \dots + (n - 1) = \frac{(1 + n - 1)(n - 1)}{2} \simeq \frac{n^2}{2}$$

$\boxed{N_c}$  Atzerakako ordezkapenaren eragiketa-kostua  $\frac{n^2}{2}$  da.



$$\text{Laburbilduz, } \boxed{N = N_a + N_b + N_c \simeq \frac{n^3}{3} + \frac{n^2}{2} + \frac{n^2}{2} \simeq \frac{n^3}{3}}$$

### Matrize bat alderantzeko

$A^{-1}$  kalkulatzeko,  $n$  sistema planteatzen dira:  $A \cdot \underline{x}_j = \underline{e}_j$ ,  $j = 1, \dots, n$ . Gai independenteak bereziak izanik, eragiketa kostua  $n \cdot n^3/3 = n^4/3$  baino askoz txikiagoa izango da. Zehazki:

**a)**  $A$  matrizea bakarrik behin triangeluarizatzen da, hau da,  $LU$  faktORIZAZIOA kalkulatu da sistema guztietan erabiltzeko. Eragiketa kostua:  $N_a = n^3/3$ .

**b)** Gai independenteak aldatzeko eragiketa-kostua:  $\underline{e}_j$  bektorean zero asko dauenez, kontuz zenbatuko ditugu: hasi baino lehen,  $j$  finkatuko dugu. Gogoratu

$$b_i^{(k)} - m_{ik} \cdot b_k^{(k)}, \quad i = k + 1, \dots, n$$

delako. Orduan,  $k = 1, \dots, j - 1$  bada, ez dira eragiketarik egiten,  $b_k^{(k)} = 0$  delako.  $k = j$  denean,  $b_k^{(k)} = 1$  denez, ez dago eragiketarik egiteko beharrik. Azkenean,  $k = j + 1, \dots, n$  denean,  $n - k$  biderkaketa egiten dira. Hori zenbatu behar dugu  $k = 1, \dots, n - 1$  denean, hau da:

$$(n - j - 1) + (n - j) + \dots + 2 + 1 = \frac{(n - j - 1)(n - j)}{2}.$$

Orduan, sistema guztiak batera kontsideratuz, eta  $i = n - j$  aldaketa eginez,

$$N_b = \sum_{j=1}^n \frac{(n - j - 1)(n - j)}{2} = \frac{1}{2} \sum_{i=n-1}^0 (i^2 - i) \simeq \frac{1}{2} \left( \frac{n^3}{3} - \frac{n^2}{2} \right) \simeq \frac{n^3}{6}.$$

**c)** Atzerakako ordezkapenak ez daukate ezaugarri berezirik. Ondorioz,  $N_c \simeq n \cdot n^2/2 = n^3/2$ .

$$\text{Laburbilduz, } \boxed{N = N_a + N_b + N_c \simeq n^3/3 + n^3/6 + n^3/2 \simeq n^3}.$$

### 2.3.2 Pibotaketarik gabeko Gauss-Jordan metodoa

Gauss metodoaren antzekoa da, baina orain,  $k$ -garren urratsean,  $k$ -garren zutabeko osagai guztiak deuseztatzen dira, diagonal nagusikoak ( $a_{kk}^{(k)}$  piboteak) izan ezik. Honela,  $n$  urrats gauzatu ondoren,  $A^{(n+1)}$  matrize diagonal bat lortzen da, atze-

rakako ordezkapena aurreztuz.

$$(A^{(k)} | \underline{b}^{(k)}) = \left( \begin{array}{cccc|ccc|c} a_{11}^{(k)} & 0 & \dots & 0 & a_{1,k}^{(k)} & \dots & a_{1,n}^{(k)} & b_1^{(k)} \\ 0 & a_{22}^{(k)} & \ddots & \vdots & a_{2,k}^{(k)} & \dots & a_{2,n}^{(k)} & b_2^{(k)} \\ 0 & 0 & \ddots & 0 & \vdots & & \vdots & \vdots \\ \vdots & & \ddots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \dots & a_{k-1,n}^{(k)} & b_{k-1}^{(k)} \\ \vdots & & & 0 & a_{k,k}^{(k)} & \dots & a_{k,n}^{(k)} & b_k^{(k)} \\ \vdots & & & \vdots & a_{k+1,k}^{(k)} & \dots & a_{k+1,n}^{(k)} & b_{k+1}^{(k)} \\ \vdots & & & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 0 & a_{n,k}^{(k)} & \dots & a_{n,n}^{(n)} & b_n^{(k)} \end{array} \right)$$

Egin behar diren aldaketak:

$$E_i^{(k+1)} = E_i^{(k)} - m_{ik} \cdot E_k^{(k)}, \quad i = 1, \dots, k-1, k+1, \dots, n,$$

non  $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$  diren. Beste erara idatzita,

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)} & \begin{cases} j = 1, \dots, k-1, \forall i \\ j = k, \dots, n, i = k. \end{cases} \\ 0 & j = k, \forall i \neq k \\ a_{ij}^{(k)} - m_{ik} \cdot a_{kj}^{(k)} & j = k+1, \dots, n \forall i \neq k \end{cases}, \quad \text{eta} \quad b_i^{(k+1)} = \begin{cases} b_i^{(k)} & i = k \\ b_i^{(k)} - m_{ik} \cdot b_k^{(k)} & i \neq k. \end{cases}$$

Azkenean, atzerakako ordezkapena egin beharrean,

$$x_i = \frac{b_i^{(n+1)}}{a_{ii}^{(n+1)}}, \quad i = 1, \dots, n$$

burutu behar da, besterik ez.

**Oharra:** etapa bat gehiago dugu, azken zutabeen ere lana baitago!

### Eragiketa-kostua

$A \cdot \underline{x} = \underline{b}$  sistema ebazteko. Gauss metodoan bezala,  $N_a$ ,  $N_b$  eta  $N_c$  kalkulatu

tugu:

- Diagonalizazioa egiteko:  $N_a \simeq n^3/2$ ;
- Bektore independentea aldatzeko:  $N_b \simeq n^2$ ;

- sistema diagonalizatu ondoren,  $\underline{x}$  askatzeko,  $N_c \simeq n$ .

Laburbilduz,  $N = N_a + N_b + N_c \simeq n^3/2$ . Nahiz eta atzerakako ordezkapena aurreztu, Gauss metodoaz ( $n^3/3$ ) baino eragiketa gehiago egiten dira.

$A^{-1}$  kalkulatzeko. Gauss-Jordan metodoa erabiliz  $A$  matrizea alderantzeko eragiketa kostua  $N \simeq n^3$  da, Gauss metodoan bezala. Hori gertatzen da atzerakako ordezkapenak aurrezteagatik.

### Memorian biltegiraketa

Sistema bat ebazten denean, Gauss metodoan bezala burutzen da biltegiraketa: biderkatzaileak gordetzen dira anulatzen diren gaien gainean, eta beste gaiak gaurkotzen dira.

Matrize bat alderantzeko, matrizean bertan biltegira daitezke egin behar diren aldatuta guztiak, bektore askeak eta guzti:

- $k = 1$  pausuan,  $\underline{e}_1 = \underline{e}_1^{(1)}$  bakarrik aldatzen da, eta  $\underline{e}_1^{(2)}$  bektore aldatuta  $A$  matrizeko lehen zutabeetan biltegira dezakegu ( $a_{11}^{(1)}$  pibotea  $n$  luzerako bektore laguntzaile batean biltegiratu ondoren).
- $k$ -garren pausuan, aldatuko dira  $\underline{e}_1^{(k)}, \dots, \underline{e}_k^{(k)}$  bektoreak, eta  $\underline{e}_1^{(k+1)}, \dots, \underline{e}_k^{(k+1)}$  bektore aldatu horiek  $A$ -ren lehenengo  $k$  zutabeetan biltegiratuko dira,  $a_{k,k}^{(k)}$  pibotea bektore laguntzailean gorde ostean.
- Diagonalizazioa amaituta, errenkada bakoitza zatitzen da dagokion pibotetik.

Laburbilduz, bakarrik behar izango dugu  $n^2 + n \simeq n^2$  leku memoria. Gauss metodoa erabiliz, aldiz,  $2n^2$  leku dira beharrezkoak, hau da, bikoitza.

### 2.3.3 Pibotaketa teknikak

Ezabapen Gaussiarra burutzerakoan aurki ditzakegu bi oztopo mota:

**Oztopo aljebraikoak:**  $a_{kk}^{(k)} = 0$  baldin bada, orduan ezin dugu zatitu!

**Zenbakizko oztopoak:**  $a_{kk}^{(k)} = \epsilon$  oso txikia bada, orduan borobiltze-errorea hedatuko da.

### Pibotaketa partzial arrunta

Aurreko oztopoak ekiditzeko, pausu bakoitzean, dagokion zutabea hautatuko dugu balio absolutu handieneko osagaia, eta hori hartuko dugu pibotetzat ezabapen Gaussiarra burutzeko. Zehazkiago, hartuko dugu pibotetzat

$$a_{l,k}^{(k)}, \quad \text{non} \quad |a_{l,k}^{(k)}| = \max\{|a_{k,k}^{(k)}|, |a_{k+1,k}^{(k)}|, \dots, |a_{n,k}^{(k)}|\}$$

betetzen den, eta errenkadak trukatu ditugu:

$$E_l^{(k)} \leftrightarrow E_k^{(k)}.$$

Trukaketa eginda, jarraituko da ezabapen Gaussiarra dakigun legez.

**2.1 oharra.** • Errenkadak trukatuz, bakarrik aldatzen da ekuazioen ordena. Hortaz, lorturiko sistema eta hasierakoa baliokideak dira.

- Batzuetan, deskribaturiko teknika ez da nahikoa borobiltze-errorea heda ez dadin (ADIBIDEAK)

### Pibotaketa partzial eskalatua

Nahiz eta pibotaketa partzial erabili, pibote bat bere errenkadako elementuekin konparatuz erlatiboki txikiak badira, borobiltze-errorea hedatu daiteke. Hori ekiditzeko, *pibotaketa partzial eskalatua* teknika erabiliko dugu, hau da:

- Errenkada bakoitzerako, *eskala-balioak* kalkulatu dira, hau da:

$$s_i = \max_{1 \leq j \leq n} |a_{ij}|, \quad i = 1, \dots, n$$

- Errenkada guztiak normalizatzen dira, hau da:  $\frac{1}{s_i} \cdot E_i$  kalkulatu dira.
- Errenkadak normalizatuta, hau da, komentaturiko arazoa gaindituta, pibotaketa partzialarekin jarrai dezakegu.

Praktikan, algoritmoa beste erara burutuko dugu:

- Praktikan, ez ditugu errenkadak eskala-balioekin zatituko, borobiltze-errorea heda ez dadin: bakarrik zatituko dugu  $k$ -garren zutabeko gaiak **piboterik hoberena aukeratzeko**. Piboterik hoberena aukeratuta, burutzen da pibotaketa, errenkadak normalizatu gabe.

- Errenkadak ez dira fisikoki trukutzen: errenkaden ordenaren eboluzioa bektore berezi batean erregistratuko dugu pibotaketa bektorean. Honela idatziko da  $k$ -garren etapan daukagun pibotaketa bektorea:

$$\underline{p}^{(k)} = (p_1^{(k)}, p_2^{(k)}, \dots, p_n^{(k)})^t$$

Hasieran, badaukagu  $\underline{p}^{(1)} = (1, 2, \dots, n)^t$ .

- Eskala-balioak bakarrik kalkulatzen dira *behin*, hasieran. Ez dira ber kalkulatu eta bakoitzean.
- **Piboteen aukeraketa.** Notazio aldetik, esango dugu lehen pibotea  $p_1$ -garren errenkadan aukeratzen dugula, bigarrena  $p_2$ -garren errenkadan, etab.  $k$ -garren etapan, aukeratu dugu jada lehenengo  $k - 1$  pibote, eta orduan, badaukagu

$$\underline{p}^{(k)} = (\overbrace{p_1, p_2, \dots, p_{k-1}}^{\text{behin-betikoak}}, p_k^{(k)}, p_{k+1}^{(k)}, \dots, p_n^{(k)})^t$$

$p_k$  pibotea aukeratzen da honako konparaketa hau eginez:

$$\max_{i \notin \{p_1, \dots, p_{k-1}\}} \frac{|a_{ik}^{(k)}|}{s_i} = \frac{|a_{lk}^{(k)}|}{s_l} \implies p_k = l$$

Orduan, pibotaketa bektorean, trukutzen dira  $l$  eta  $p_k^{(k)}$  zenbakien posizioak. Azkenean, izango dugu  $\underline{p} = (p_1, \dots, p_n)^t$ .

**2.1 adibidea.** Demagun  $n = 7$  eta  $p_1 = 2, p_2 = 3$  eta  $p_3 = 7$  aukeratu ditugula.

Orduan, badaukagu  $\underline{p}^{(4)} = (\overbrace{2, 3, 7}^{\text{behin-betikoak}}, 4, 5, 6, 1)^t$ .  $p_4$  aukeratzeko, kalkulatu genuke honako maximo hau:

$$\max \left\{ \frac{|a_{14}^{(4)}|}{s_1}, \frac{|a_{44}^{(4)}|}{s_4}, \frac{|a_{54}^{(4)}|}{s_5}, \frac{|a_{64}^{(4)}|}{s_6} \right\}$$

Demagun aurreko maximoa  $\frac{|a_{64}^{(4)}|}{s_6}$  dela. Orduan, izango genuke  $p_4 = 6$ , eta pibotaketa bektorea honela aldatuko da:

$$\underline{p}^{(5)} = (\overbrace{2, 3, 7, 6}^{\text{behin-betikoak}}, 5, 4, 1)^t$$

### Interpretazio matriziala

Gauss metodoa erabiltzen badugu pibotaketa partzial eskalatuaz (edo eskalatu gabe), orduan ezabapen Gaussiarra amaitu ondoren, eraiki ditzakegu  $U$  eta  $L$  matrizeak, errenkadak eta biderkatzaileak pibotaketa bektoreaz berrordenatuz. Baina honela, ez dugu lortzen jatorriko  $A$  matrizearen  $LU$  faktORIZAZIORIK, baizik eta bere berrordenazio baten faktORIZAZIOA. Zehazkiago, izango genuke:

$$L \cdot U = P^t \cdot A, \quad \text{non } P = (\underline{e}_{p_1}, \underline{e}_{p_2}, \dots, \underline{e}_{p_m}) \text{ dugun.}$$

Prozesua burutzeko, behar ditugu bi bektore laguntzaile:  $\underline{p}$  eta  $\underline{s} = (s_1, \dots, s_n)^t$ .

[ADIBIDEAK]

### Egonkortasun teoremak

Ezabapen Gaussiarra erabiltzen dugunean, bi kasu berezietan ez da beharrezkoa pibotaketa erabiltzea.

**2.4 definizioa.** Esango dugu  $A \in \text{Mat}(n \times n, \mathbb{R})$  *diagonal hertsiki menperatzaileduna* dela baldin eta honako baldintza hau betetzen bada:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, \dots, n$$

**2.5 definizioa.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$  matrize simetriko bat. Esango dugu  $A$  *positiboki definitua* dela baldin eta

$$\underline{x}^t \cdot A \cdot \underline{x} > 0 \text{ bada } \forall \underline{0} \neq \underline{x} \in \mathbb{R}^n.$$

**2.4 teorema.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$ .  $A$  *diagonal hertsiki menperatzaileduna* bada, orduan  $A \cdot \underline{x} = \underline{b}$  erako edozein sistema ebaz daiteke pibotaketarik gabeko ezabapen Gaussiarreko metodoak erabiliz (Gauss zein Gauss-Jordan). Are gehiago, kalkuluak egonkorak izango dira.

**2.5 teorema.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$  matrize simetriko bat. Orduan,  $A$  *positiboki definitua* bada, orduan  $A \cdot \underline{x} = \underline{b}$  erako edozein sistema ebaz daiteke pibotaketarik gabeko ezabapen Gaussiarreko metodoak erabiliz (Gauss zein Gauss-Jordan). Are gehiago, kalkuluak egonkorak izango dira.

## 2.4 Ezabapen metodo trinkoak

### 2.4.1 Sarrera

Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$ , eta bitez  $L, U \in \text{Mat}(n \times n, \mathbb{R})$  matrize triangeluarrak (behe- eta goi-triangeluarrak, hurrenez hurren), zeintzutarako  $A = L \cdot U$  faktorizazioa dugun. Orduan, esango dugu  $A$  matrizeak  $LU$  faktorizazio bat onartzen duela.

#### Sistemak ebazteko teknika trinkoa

$A$  matrize bat faktorizatuta badaukagu, orduan,  $A \cdot \underline{x} = \underline{b}$  sistema ebaz dezakegu nahiko erraz, honako teknika hau erabiliz:

- $A \cdot \underline{x} = \underline{b} \implies (L \cdot U) \cdot \underline{x} = \underline{b} \implies L \cdot (U \cdot \underline{x}) = \underline{b}$  daukagu.  $(U \cdot \underline{x}) = \underline{y}$  aldagai aldaketa eginez, badaukagu jatorriko sistema planteatu daiteke bi sistemetan:

$$L \cdot \underline{y} = \underline{b} \quad \text{eta} \quad U \cdot \underline{x} = \underline{y}$$

- $L \cdot \underline{y} = \underline{b}$  sistema ebaz daiteke aurrerakako ordezkapenaz (atzerakako ordezkapenaren algoritmo berdina, baino aurrerraka definituta).
- $\underline{y}$  lortuta,  $U \cdot \underline{x} = \underline{y}$  sistema ebazten da atzerakako ordezkapenaz,  $\underline{x}$  soluzioa lortuz.

#### Matrizeak alderantzeko teknika trinkoak

$A = L \cdot U$  faktorizazioa aprobetxatu daiteke, ere, alderantzizkoa kalkulatzeko hurrengo bi erataraz:

- $A \cdot \underline{x} = L \cdot (U \cdot \underline{x}) = \underline{e}_i$  sistema bakoitza ebaz daiteke goian aipaturiko teknikaz.
- Aljebra pixkat erabiliz, badaukagu  $A^{-1} = (L \cdot U)^{-1} = U^{-1} \cdot L^{-1}$ . Orduan, kalkula daitezke, aparteaz,  $L^{-1}$  eta  $U^{-1}$  (Gauss-Jordan metodoaz, edo dagokion ordezkapena erabiliz), eta gero biderkatu ordena egokian  $A^{-1}$  lortzeko.

### *LU* faktORIZAZIOAREN BAKARTASUNA

Nabaritu badaudela *LU* faktorizaziorik onartzen ez duten matrizeak; honako hauek, besteak beste:

$$\begin{pmatrix} 0 & 1 \\ 1 & x \end{pmatrix} \quad \text{eta} \quad \begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & & \dots & \\ \vdots & & & \\ 1 & & \dots & \end{pmatrix}$$

Bestalde, nabaritu Gauss metodoa burutu badaiteke (adibidez, 2.4 edo 2.5 teoremaren baldintza betetzen bada), orduan, *A*-ren *LU* faktorizazio bat lortzen dela. Nabaritu, ere, *LU* faktorizaziorik existitzekotan, faktorizazio horrek ez daukala zertan izan behar bakarra, honako adibide simple honetan ikus daitekeenez:

**2.2 adibidea.** Bedi *A* matrize diagonal bat. Matrize diagonal guztiak goi- eta behe-triangeluar izanik, badaukagu  $L = A, U = I_n$  eta  $L = I_n, U = A$  faktorizazioak *A*-ren bi *LU* faktorizazio posible direla.

Are gehiago, bedi orain *A* edozein matrize karratu. Orduan, *LU* faktorizazio bat emanda, beti aurki ditzakegu beste *LU* faktorizazio desberdinak (beraz, infinituak) honako truko hau erabiliz: bedi  $D \neq I_n$  matrize diagonal ez-singular bat. Orduan,

$$A = L \cdot U = L \cdot (D \cdot D^{-1}) \cdot U = \overbrace{(L \cdot D)}^{=L_1} \cdot \overbrace{(D^{-1} \cdot U)}^{=U_1} = L_1 \cdot U_1$$

Bestalde, *A*-ren *LU* faktorizazio bat ezaguturik, eraiki ditzakegu bere *LU* faktorizazio guztiak aurreko trukoa erabiliz, honako lema honetan ikus dezakegunez:

**2.6 lema.** *Matrize baten edozein bi LU faktorizazio elkaturik daude matrize diagonal baten bidez honako zentzu honetan:  $A = L_1 \cdot U_1 = L_2 \cdot U_2$  badugu, orduan existitzen da *D* matrize diagonal ez-singular bat zeinerako*

$$L_1 = L_2 \cdot D \quad \text{eta} \quad U_1 = D^{-1} \cdot U_2 \quad \text{baitira.} \quad (2.1)$$

*Frogapena.*

$$L_1 \cdot U_1 = A = L_2 \cdot U_2 \implies \overbrace{L_2^{-1} \cdot L_1}^{\text{behe-triang.}} = \overbrace{U_2 \cdot U_1^{-1}}^{\text{goi-triang.}} = D \quad (\text{goi+behe} \implies \text{diag}),$$

eta hortik, zuzenean, lortzen dira (2.1)





### 2.4.2 Aurrerakako ordezkapena

Bedi  $L \in \text{Mat}(n \times n, \mathbb{R})$  behe-triangeluarra, eta hartu aintzakotzat  $L \cdot \underline{x} = \underline{b}$  sistema:

$$\begin{cases} l_{1,1} \cdot x_1 & & & & & = b_1 \\ l_{1,2} \cdot x_1 & + l_{2,2} \cdot x_2 & & & & = b_2 \\ \vdots & & \ddots & & & = \vdots \\ l_{n-1,1} \cdot x_1 & + l_{n-1,2} \cdot x_2 & + \dots & & & = b_{n-1} \\ l_{n,1} \cdot x_1 & l_{n,2} \cdot x_2 & + \dots & + l_{n,n-1} \cdot x_{n-1} & + l_{n,n} \cdot x_n & = b_n \end{cases}$$

Sistema hori ebazteko, *aurrerakako ordezkapena* algoritmoa erabiliko dugu, honela deskribatzen dena:

$$\begin{cases} x_1 = \frac{b_1}{l_{1,1}} \\ x_k = \frac{\left( b_k - \sum_{j=1}^{k-1} l_{k,j} \cdot x_j \right)}{l_{k,k}} \quad k = 2, \dots, n \end{cases}$$

Nabaritu eragiketa kostua atzerakako ordezkapenarena dela:  $N \simeq n^2/2$  (frogatzen da 2.2 atalean bezala).

### 2.4.3 Doolittle eta Crout algoritmoak

$LU$  faktORIZAZIO BAT lortzeko estrategia orokorra  $A = L \cdot U$  ekuazio matritziala planteatzean datza:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & & & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & \dots & \dots & 0 \\ l_{21} & l_{22} & \ddots & & 0 \\ l_{31} & l_{32} & l_{33} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ l_{n1} & l_{n2} & \dots & \dots & l_{nn} \end{pmatrix} \cdot \begin{pmatrix} u_{11} & u_{12} & \dots & \dots & u_{1n} \\ 0 & u_{22} & \dots & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \vdots & & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & u_{nn} \end{pmatrix}$$

Biderkaketa burutzean,  $A$  matrizeko osagai bakoitzerako planteatzen da ekuazio bat:

$$a_{ij} = \sum_{r=1}^n l_{ir} \cdot u_{rj}, \quad i, j \in \{1, \dots, n\}$$



**Doolittle:** Inposatzen baditugu  $l_{kk} = 1$ ,  $k = 1, \dots, n$  baldintzak (2.2) ekuazioetan, eta goiko ordena erabiltzen badugu, lortzen dugu honako algoritmo hau, Doolittle algoritmoa deritzona:

$$k = 1, \dots, n \left\{ \begin{array}{l} u_{k,k} = a_{k,k} - \sum_{r=1}^{k-1} l_{k,r} \cdot u_{r,k} \\ u_{k,j} = a_{k,j} - \sum_{r=1}^{k-1} l_{k,r} \cdot u_{r,j} \quad j = k+1, \dots, n \\ l_{i,k} = \frac{1}{u_{k,k}} \left( a_{i,k} - \sum_{r=1}^{k-1} l_{i,r} \cdot u_{r,k} \right) \quad i = k+1, \dots, n \end{array} \right.$$

**Crout:** Inposatzen baditugu  $u_{kk} = 1$ ,  $k = 1, \dots, n$  baldintzak (2.2) ekuazioetan, eta goiko ordena erabiltzen badugu, lortzen dugu honako algoritmo hau, Crout algoritmoa deritzona:

$$k = 1, \dots, n \left\{ \begin{array}{l} l_{k,k} = a_{k,k} - \sum_{r=1}^{k-1} l_{k,r} \cdot u_{r,k} \\ u_{k,j} = \frac{1}{l_{k,k}} \left( a_{k,j} - \sum_{r=1}^{k-1} l_{k,r} \cdot u_{r,j} \right) \quad j = k+1, \dots, n \\ l_{i,k} = a_{i,k} - \sum_{r=1}^{k-1} l_{i,r} \cdot u_{r,k} \quad i = k+1, \dots, n \end{array} \right.$$

[ADIBIDEAK]

**2.2 oharrak.** • Doolittle metodoaz lortzen den  $LU$  faktORIZAZIOA eta Gauss metodoaz lortzen dena berdina dira ( $l_{kk} = 1$ ,  $\forall k$ ).

- 2.4 edo 2.5 teoremaren baldintza betetzen bada, orduan, deskribaturiko Crout zein Doolittle algoritmoak egonkorak dira  $LU$  faktORIZAZIO bat lortzeko.
- Pibotaketa bateragarria da Crout eta Doolittle algoritmoekin (hala ere, ikasgai honetan, guk ez dugu erabiliko).
- FaktORIZAZIO trinkoko algoritmoetan honako azdierazpen hauek agertzen dira ezezagun bakoitza kalkulatzeko:

$$\frac{\sum_r \alpha_r \cdot \beta_r}{\gamma}$$

Adierazpen horiek zenbakizko abantaila bat dute zehaztasun sinplez lanegiten denean: memoriako aldagai bat dimentsiona dezakegu zehaztasun bikoitzez, non kalkula daitekeen zatikiaren zenbakitzailea zehaztasun bikoitzez. Gero, zehaztasun sinpleko  $\gamma$  osagaitik zatituz, zehaztasun sinplera igaroko dugu. Emaitza  $LU$  faktORIZAZIOKO behin betiko gai bat izango da, eta nahiz eta zehaztasun sinplez adierazita izan, azken eragiketa arte zehaztasun bikoitzez kalkulatu izan da. Ezabapen Gaussiarreko metodoetaz, aldiz, etapaz etapako prozesuak izanik, zehaztasun (ia) bikoitza mantentzeko era bakarra matrize osoa zehaztasun bikoitzez dimentsionatzea da. Ezabapen trinkoaz, memoriako leku kopuru erdia erabiliz egin daiteke.

### Memorian biltegiraketa

Algoritmoaren deskribapenean ikus daiteke  $a_{i,j}$  osagai bakoitza behin bakarrik erabiltzen dela,  $L$  edo  $U$  matrizeko gai bat kalkulatzeko. Orduan,  $a_{i,j}$  elementu hori ezabatuko dugu, eta hutsik geratzen den lekuan gordeko da kalkulaturiko elementu berria. Adibidez, Doolittle algoritmoaz honako ordezkapen hauek egiten dira:

$$u_{k,k} \mapsto a_{k,k} \quad l_{i,k} \mapsto a_{i,k} \quad u_{k,j} \mapsto a_{k,j}.$$

Azkenean, Gauss metodoan bezala,  $\begin{pmatrix} \diagdown & U \\ L & \diagdown \end{pmatrix}$  erako matrize bat lortzen da. Sistema bat ebazteko, aurrerakako ordezkapena egiten da  $\underline{y}$  bektorea  $\underline{b}$  gainean idatziz, goitik behera. Gero, atzerakako ordezkapena burutzen da (behetik gora),  $\underline{x}$  soluzioa  $\underline{y}$  gainean idatziz.

### Eragiketa Kostua

**Sistemaren ebazpena.** Ezabapen Gaussiarrean bezala, zatituko dugu eragiketa kostua 3 prozesutan:

- $A = L \cdot U$  faktORIZAZIOA lortzeko eragiketa kostua  $N_a \simeq n^3/3$  (Crout edo Doolittle metodoaz);
- $L \cdot \underline{y} = \underline{b}$  sistemaren ebazpena (aurrerakako ordezkapen bat):  $N_b \simeq n^2/2$ .
- $U \cdot \underline{x} = \underline{y}$  sistemaren ebazpena (atzerakako ordezkapen bat):  $N_c \simeq n^2/2$ .

Guztira,  $N = N_a + N_b + N_c \simeq n^3/3 + n^2 \simeq n^3/3$ .

**Alderantzizkoaren kalkulua.** Alderantzizkoa kalkulatzeko, 2.4.1 atalean planteaturiko edozein bi tekniketaz, eragiketa kostua berdina da:  $N \simeq n^3$  ( $LU$  faktORIZAZIOA barne, jakina, Crout zein Doolittle algoritmoaz).

### 2.4.4 Cholesky metodoa

A matrizea simetrikoa eta positiboki definitua bada, orduan  $LU$  faktORIZAZIO berezi bat lor dezakegu  $U = L^t$  hartuz. Metodoa honako teorema honetan dago oinarrituta.

**2.7 teorema.** *A matrizea simetrikoa eta positiboki definitua da baldin eta soilik baldin  $L = (l_{ij})$  matrize behe-triangeluar bat existitzen bada, zeinen diagonal nagusiko elementuak positiboak baitira ( $l_{ii} > 0, i = 1, \dots, n$ ),  $A = L \cdot L^t$  betetzen delarik.*

Cholesky metodoa burutzeko, planteatzen dira (2.2) ekuazioak  $U = L^t$  suposatuz. Algoritmoaren deskribapena honako hau da:

$$k = 1, \dots, n, \quad \begin{cases} l_{k,k} = \sqrt{a_{k,k} - \sum_{r=1}^{k-1} (l_{k,r})^2} \\ l_{i,k} = \frac{1}{l_{k,k}} \left( a_{i,k} - \sum_{r=1}^{k-1} l_{i,r} \cdot l_{k,r} \right) \quad i = k + 1, \dots, n. \end{cases}$$

**2.1 oharra.** Metodoa test moduan erabili ohi da:  $A$  matrize simetrikoa badugu, orduan saia gaitzke Cholesky algoritmoa erabiltzen: posible bada (hau da, prozesuan sortzen diren erro karratuak kalkulagarriak badira; zehazkiago,  $a_{k,k} > \sum_{r=1}^{k-1} (l_{k,r})^2$  betetzen badira), orduan, matrizea positiboki definitua izango da. Bestela, matrizea ez da positiboki definitua izango.

### Eragiketa Kostua

Erro karratu bat kalkulatzeko eragiketa kostua  $\lambda$  biderketakoa da, non  $3 \leq \lambda \leq 10$ , gutxi gora-behera.

Sistema bat ebazteko eragiketa kostua  $N \simeq n^3/6 + \lambda n \simeq n^3/6$  da. Matrize bat alderantzeko, aljebra pixkat erabiliz, prozesua izugarri sinplifika daiteke:

$$A^{-1} = (L \cdot L^t)^{-1} = (L^t)^{-1} \cdot L^{-1} = (L^{-1})^t \cdot L^{-1}.$$

Honela,  $A^{-1}$  kalkulatzeko eragiketa kostua  $N \simeq n^3/2$  da.

[ADIBIDEA]

### 2.4.5 Jacobiren matrize tridiagonalak

**2.6 definizioa.** Honako itxura honetako matrize bati *Jacobi matrize tridiagonal* deritzo:

$$\begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_2 & a_2 & c_2 & 0 & \dots & 0 \\ 0 & b_3 & a_3 & c_3 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & & 0 & b_{n-1} & a_{n-1} & c_{n-1} \\ 0 & \dots & & & 0 & b_n & a_n \end{pmatrix}$$

**2.8 teorema.** Baldin  $A$  Jacobi matrize tridiagonal bat bada, eta gainera honako baldintza hauek betetzen badira:

- i)  $|a_1| > |c_1| > 0$ ;
- ii)  $|a_i| \geq |b_i| + |c_i|$  eta  $b_i \cdot c_i \neq 0 \quad i = 2, \dots, n-1$ ;
- iii)  $|a_n| > |b_n| > 0$ ,

orduan,  $A$  matrizeak honako LU faktORIZAZIO hauek onartzen ditu:

#### Crout

$$A = LU = \begin{pmatrix} \alpha_1 & 0 & \dots & 0 \\ b_2 & \alpha_2 & 0 & \dots & 0 \\ 0 & b_3 & \alpha_3 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & b_{n-1} & \alpha_{n-1} & 0 \\ 0 & \dots & & 0 & b_n & \alpha_n \end{pmatrix} \cdot \begin{pmatrix} 1 & \gamma_1 & 0 & \dots & 0 \\ 0 & 1 & \gamma_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & & 0 & 1 & \gamma_{n-1} \\ 0 & \dots & \dots & & 0 & 1 \end{pmatrix}$$

#### Doolittle

$$A = LU = \begin{pmatrix} 1 & 0 & \dots & 0 \\ \beta_2 & 1 & 0 & \dots & 0 \\ 0 & \beta_3 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & \beta_{n-1} & 1 & 0 \\ 0 & \dots & & 0 & \beta_n & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 & c_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & c_2 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & & 0 & \alpha_{n-1} & c_{n-1} \\ 0 & \dots & \dots & & 0 & \alpha_n \end{pmatrix}$$

Gainera, algoritmoa egonkorra izango da.

[ADIBIDEA]

**Algoritmoa**

Crout erakoa (Doolittle erakoa ariketatzat geratuko da):

- $\alpha_1 = a_1, \quad \gamma_1 = c_1/\alpha_1;$
- $\alpha_i = a_i - b_i \cdot \gamma_{i-1}, \quad \gamma_i = c_i/\alpha_i, \quad i = 2, \dots, n.$

**Eragiketa kostua**

Sistema bat ebazteko:  $N = N_a + N_b + N_c = 2n + 2n + n = 5n.$

Alderantzizkoa kalkulatzeko:  $N \simeq 2n^2$

**Memorian biltegiraketa**

$A$  matrizea 3 bektoretan gordetzen da,  $n$  luzerakoak:  $\underline{a}, \underline{b}$  eta  $\underline{c}$ . Hiru bektore horiek nahikoak dira  $LU$  faktORIZAZIOA gordetzeko: Adibidez, Doolittle metodoa erabiltzen bada, ordezkaturako genituzke  $\alpha_i \mapsto a_i$  eta  $\beta_i \mapsto b_i$  pausu bakoitzean.

## 2.5 Errorea eta Arazoaren Baldintzaketa

### 2.5.1 Norma bektorialak eta matrizialak. Norma naturalak

**2.7 definizioa.**  $\mathbb{R}^n$  espazioan definituriko norma bektorial bat

$$\begin{aligned} \|\cdot\| : \mathbb{R}^n &\longrightarrow \mathbb{R} \\ \underline{x} &\longmapsto \|\underline{x}\| \end{aligned}$$

aplikazio bat da, honako propietate hauek betetzen dituen:

- i)  $\|\underline{x}\| \geq 0 \quad \forall \underline{x} \in \mathbb{R}^n$ ;
- ii)  $\|\underline{x}\| = 0 \iff \underline{x} = \underline{0}$ ;
- iii)  $\|\alpha \cdot \underline{x}\| = |\alpha| \cdot \|\underline{x}\|, \quad \forall \underline{x} \in \mathbb{R}^n, \forall \alpha \in \mathbb{R}$ ;
- iv)  $\|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\|, \quad \forall \underline{x} + \underline{y} \in \mathbb{R}^n$ .

**Adibideak:**

- $\|\underline{x}\|_1 = \sum_{i=1}^n |x_i|$  normari  $l_1$  norma esaten zaio.
- $\|\underline{x}\|_2 = \sqrt{x_1^2 + \dots + x_n^2}$  normari  $l_2$  norma edo norma Euklidiarra esaten zaio.
- $\|\underline{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$  normari  $l_\infty$  norma edo maximoaren norma esaten zaio.

**2.8 definizioa.** Esango dugu  $\|\cdot\|$  eta  $\|\cdot\|'$  normak baliokideak direla baldin eta existitzen badira  $C_1, C_2$  konstante erreal positiboak zeintzuetarako

$$C_1 \cdot \|\underline{x}\| \leq \|\underline{x}\|' \leq C_2 \cdot \|\underline{x}\| \quad \forall \underline{x} \in \mathbb{R}^n$$

betetzen den.

**2.2 oharra.**  $\mathbb{R}^n$  espazioko norma bektorial guztiak baliokideak dira.

**2.9 definizioa.** Esango dugu  $\{\underline{x}^{(k)}\}_{k \geq 1}$  bektoreen segidak  $\underline{x}$  bektorerantz konbergitzen duela baldin eta  $\|\underline{x}^{(k)} - \underline{x}\| \xrightarrow{k \rightarrow \infty} 0$  betetzen bada.



**2.9 teorema.**  $\{\underline{x}^{(k)}\}_{k \geq 1}$  segidak  $l_\infty$  normarekiko  $\underline{x}$  bektorerantz konbergitzen du baldin eta soilik baldin

$$\lim_{k \rightarrow \infty} x_i^{(k)} = x_i \quad i = 1, 2, \dots, n$$

bada.

**2.3 oharra.**  $\mathbb{R}^n$ -ko norma bektorial guztiak baliokideak direnez, segida bat  $l_\infty$  normarekiko konbergentea baldin bada, orduan ondoriozta dezakegu  $l_1$  eta  $l_2$  normekiko konbergentea izango dela (baita edozein normarekin ere).

**2.10 definizioa.** *Norma matrizial* bat

$$\begin{aligned} \|\cdot\| : \text{Mat}(n \times n, \mathbb{R}) &\longrightarrow \mathbb{R} \\ A &\longmapsto \|A\| \end{aligned}$$

aplikazio bat da, honako propietate hauek betetzen dituena:

- i)  $\|A\| \geq 0, \quad \forall A \in \text{Mat}(n \times n, \mathbb{R})$
- ii)  $\|A\| = 0 \iff A = 0$  matrize nulua;
- iii)  $\|\alpha \cdot A\| = |\alpha| \cdot \|A\|, \quad \forall A \in \text{Mat}(n \times n, \mathbb{R}), \forall \alpha \in \mathbb{R};$
- iv)  $\|A + B\| \leq \|A\| + \|B\|, \quad \forall A, B \in \text{Mat}(n \times n, \mathbb{R});$
- iv)  $\|A \cdot B\| \leq \|A\| \cdot \|B\|, \quad \forall A, B \in \text{Mat}(n \times n, \mathbb{R}).$

**2.10 teorema.**  $\mathbb{R}^n$  espazioko  $\|\cdot\|$  norma bektorial bat emanda, orduan  $\|\cdot\|$  norma matrizial bat defini dezakegu honako formula honetaz:

$$\|A\| = \max_{\|\underline{x}\|=1} \|A \cdot \underline{x}\|.$$

Norma matrizial hori *norma natural* bat dela esango dugu,  $\|\cdot\|$  norma bektorialari dagokiona.  $A$  matrize baten norma naturalak neurtzen du 1 erradioko esferaren deformazioa  $A: \mathbb{R}^n \longrightarrow \mathbb{R}^n$  aplikazio linealaren bidez.

[IRUDIAK  $\|\cdot\|_2$  normarako]

**2.11 teorema.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$ . Orduan, badaukagu:

- $\|A\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$
- $\|A\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$

$$\bullet \|A\|_2 = \sqrt{\rho(A^t \cdot A)}$$

**2.1 definizioak.**  $A \in \text{Mat}(n \times n, \mathbb{R})$  bada eta  $A \cdot \underline{x} = \lambda \underline{x}$  bada, orduan esango dugu  $\lambda$  balioa  $A$ -ren *autobalio bat* dela eta  $\underline{x}$  autobalio hori dagokion *autobektore bat* dela esango dugu.  $\lambda$  autobalioa erreal edo konplexu izan daiteke, segun eta  $\underline{x} \in \mathbb{R}^n$  edo  $\underline{x} \in \mathbb{C}^n$  dagoen. Matrizearen autobalioen multzoa  $\text{Spec}(A)$  adierazpenaz idatziko dugu.  $A$  matrizearen *erradio espektrala* holan definitzen da:

$$\rho(A) = \max_{\lambda \in \text{Spec}(A)} |\lambda|$$

**2.12 teorema.** *Edozein norma naturalarako, honako propietate hauek betetzen dira:*

$$i) \|A \cdot \underline{y}\| \leq \|A\| \cdot \|\underline{y}\|, \quad \forall \underline{y} \in \mathbb{R}^n;$$

$$ii) \rho(A) \leq \|A\|.$$

*Frogapena:*

i)

$\underline{y} = \underline{0}$  bektore nulua bada, orduan, propietatea betetzen da, zuzenean ( $A \cdot \underline{0} = \underline{0}$ .)

$\underline{y} \neq \underline{0}$  bada, orduan normaliza dezakegu bektorea:

$$\underline{v} = \frac{\underline{y}}{\|\underline{y}\|}.$$

Nabaritu  $\|\underline{v}\| = 1$  dela. Orain, norma natulararen definizioa aplikatuz:

$$\|A\| = \max_{\|\underline{x}\|=1} \|A \cdot \underline{x}\| \geq \|A \cdot \underline{v}\| = \left\| A \cdot \frac{\underline{y}}{\|\underline{y}\|} \right\| = \frac{\|A \cdot \underline{y}\|}{\|\underline{y}\|}.$$

Frogatu dugu  $\|A\| \geq \frac{\|A \cdot \underline{y}\|}{\|\underline{y}\|}$  dela. Ondorioz,  $\|A \cdot \underline{y}\| \leq \|A\| \cdot \|\underline{y}\|$ .

ii) Bedi  $\lambda \in \text{Spec}(A)$ , eta bedi  $\underline{x}$ ,  $\lambda$ -ri dagokion autobektore bat. Orduan,  $A \cdot \underline{x} = \lambda \underline{x}$  daukagu, eta orduan, normak hartuz eta  $i)$  propietatea aplikatuz,

$$|\lambda| \|\underline{x}\| = \|A \cdot \underline{x}\| \leq \|A\| \cdot \|\underline{x}\| \implies |\lambda| \leq \|A\|, \forall \lambda \in \text{Spec}(A).$$

Amaitzeko,

$$\rho(A) = \max_{\lambda \in \text{Spec}(A)} |\lambda| \leq \|A\|.$$



### 2.5.2 Errorearen azterketa

Norma bektorialeiei esker, defini dezakegu bektoreen arteko distantzia, eta hortik, errorea.

**2.2 definizioak.**  $\mathbb{R}^n$  espazioko  $\underline{x}$  bektorea ebazpen zehatza bada eta  $\underline{\tilde{x}}$  bere hurbilpen bat bada, orduan

$$\underline{e} = \underline{x} - \underline{\tilde{x}}$$

*errore bektorea* dela esango dugu.  $\mathbb{R}^n$  espazioko  $\|\cdot\|$  norma bat emanda, *errore absolutua*

$$e = \|\underline{e}\| = \|\underline{x} - \underline{\tilde{x}}\|$$

izango da, eta *errore erlatiboa*

$$e_r = \frac{e}{\|\underline{x}\|} = \frac{\|\underline{x} - \underline{\tilde{x}}\|}{\|\underline{x}\|}$$

izango da. (Batzutan,  $e_r = \frac{\|\underline{x} - \underline{\tilde{x}}\|}{\|\underline{\tilde{x}}\|}$  erabiltzen da.)

Demagun, orain,  $A \cdot \underline{x} = \underline{b}$  sistemarako  $\underline{\tilde{x}}$  soluzio hurbildua lortu egin dugula.  $\underline{e} = \underline{x} - \underline{\tilde{x}}$  errore bektorea dugu, baina errorea neurtzeko beste era ikusiko dugu

**2.3 definizioak.** Aurreko baldintzetan:

- $\underline{r} = \underline{b} - A \cdot \underline{\tilde{x}}$  bektoreari *hondar errore* bektorea deritzo;
- $\|\underline{r}\|$  balioari *hondar errore* deritzo;
- $\frac{\|\underline{r}\|}{\|\underline{b}\|}$  balioari *hondar errore erlatibo* deritzo.

Nabaritu  $\underline{r} = \underline{b} - A \cdot \underline{\tilde{x}} = A \cdot \underline{x} - A \cdot \underline{\tilde{x}} = A \cdot (\underline{x} - \underline{\tilde{x}}) = A \cdot \underline{e}$  dela. Hondar erroreak deskribatzen du nola betetzen duen sistema soluzio hurbilduak. Azken hori adierazgarria izan daiteke kasu batzutan, baina beste batzuetan, aldiz, engainagarria.

[ADIBIDEA]

**2.13 teorema.** Bedi  $A \in \text{Mat}(n \times n, \mathbb{R})$  matrize ez-singular bat, eta demagun soluzio hurbildu bat lortu dugula,  $\underline{e}$  errorea eta  $\underline{r}$  hondar errorea sortuz. Orduan,  $\|\cdot\|$  edozein norma naturaltarako, badaukagu:

$$i) \frac{\|\underline{r}\|}{\|A\|} \leq \|\underline{e}\| \leq \|A^{-1}\| \cdot \|\underline{r}\|;$$

$$ii) \frac{1}{\|A\| \cdot \|A^{-1}\|} \frac{\|\underline{r}\|}{\|\underline{b}\|} \leq \frac{\|\underline{e}\|}{\|\underline{x}\|} \leq \|A\| \cdot \|A^{-1}\| \frac{\|\underline{r}\|}{\|\underline{b}\|}.$$

Frogapena:

$i)$

$$\underline{r} = A \cdot \underline{e} \implies \|\underline{r}\| = \|A \cdot \underline{e}\| \leq \|A\| \cdot \|\underline{e}\| \implies \frac{\|\underline{r}\|}{\|A\|} \leq \|\underline{e}\|.$$

Bestalde,

$$\underline{r} = A \cdot \underline{e} \implies A^{-1} \cdot \underline{r} = A^{-1} \cdot A \cdot \underline{e} = \underline{e} \implies \|\underline{e}\| = \|A^{-1} \cdot \underline{r}\| \leq \|A^{-1}\| \cdot \|\underline{r}\| \implies i)$$

$ii)$

Alde batetik,

$$\underline{b} = A \cdot \underline{x} \implies \|\underline{b}\| = \|A \cdot \underline{x}\| \leq \|A\| \cdot \|\underline{x}\| \implies \frac{\|\underline{b}\|}{\|A\|} \leq \|\underline{x}\|.$$

Bestalde,

$$\underline{b} = A \cdot \underline{x} \implies A^{-1} \cdot \underline{b} = A^{-1} \cdot A \cdot \underline{x} = \underline{x} \implies \|\underline{x}\| = \|A^{-1} \cdot \underline{b}\| \leq \|A^{-1}\| \cdot \|\underline{b}\|$$

Ondorioz, frogatu dugu

$$\frac{\|\underline{b}\|}{\|A\|} \leq \|\underline{x}\| \leq \|A^{-1}\| \cdot \|\underline{b}\|$$

dela. Amaitzeko,  $i)$  formula azken horrekin zatituz,  $ii)$  lortzen da. ♣

**2.11 definizioa.**  $\|A\| \cdot \|A^{-1}\|$  balioari  $A$  matrizearen *baldintzatze-zenbakia* edo *egokitze-zenbakia* deritzo, eta  $\mathcal{K}(A)$  notazioaz idatziko dugu.

Beraz, azken teoremako bigarren formula berridatz dezakegu honela:

$$\frac{1}{\mathcal{K}(A)} \frac{\|\underline{r}\|}{\|\underline{b}\|} \leq \frac{\|\underline{e}\|}{\|\underline{x}\|} \leq \mathcal{K}(A) \frac{\|\underline{r}\|}{\|\underline{b}\|}$$

**2.3 oharra.** •  $\mathcal{K}(A) \simeq 1$  bada, orduan, errore erlatiboa eta hondar errorea tamainu berekoak izango dira. Kasu horietan esan dezakegu arazoa ondo baldintzatua edo ondo egokituta dagoela.

- $\mathcal{K}(A) \gg 1$  bada, orduan hondar errorea ez da fidagarria izango.

Nahiz eta  $\mathcal{K}(A)$  normaren menpeko balioa izan, honako emaitza hau daukagu:

**2.14 proposizioa.**  $\mathcal{K}(A) \geq 1, \quad \forall A \in \mathcal{M}_n, \text{ norma natural orotarako.}$

*Frogapena:*

$$\text{Alde batetik, } \|I\| = \max_{\|x\|=1} \|I \cdot x\| = \max_{\|x\|=1} \|x\| = \max_{\|x\|=1} 1 = 1.$$

$$\text{Bestalde, } 1 = \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \mathcal{K}(A). \quad \clubsuit$$

### Sistema Perturbatuak

Batzutan, nahiz eta teorikoki  $A \cdot x = b$  sistema ebatzi nahi izan, praktikan, hasierako erroreak edo izandako perturbazio bat dela eta, sistema perturbatu bat ebazten da:

$$\hat{A} \cdot \hat{x} = b,$$

non  $\hat{A}$  matrize perturbatu bat den.  $\delta(A) = A - \hat{A}$  matrizea osagaien erroreek osatua da.

**2.15 proposizioa.** *Aurreko baldintzetan, honako formula hau betetzen da:*

$$e_r = \frac{\|x - \hat{x}\|}{\|\hat{x}\|} \leq \mathcal{K}(A) \cdot \frac{\|\delta(A)\|}{\|A\|}$$

*Frogapena:* Badaukagu  $A \cdot (x - \hat{x}) = A \cdot x - (\hat{A} + \delta A) \cdot \hat{x} = b - b - \delta A \cdot \hat{x}$ .

Hortaz,  $x - \hat{x} = -A^{-1} \cdot \delta A \cdot \hat{x} \implies \|x - \hat{x}\| = \|A^{-1} \cdot \delta A \cdot \hat{x}\| \leq \|A^{-1}\| \cdot \|\delta A\| \cdot \|\hat{x}\|$ , eta, amaitzeko,

$$e_r = \frac{\|x - \hat{x}\|}{\|\hat{x}\|} \leq \|A^{-1}\| \cdot \|\delta A\| = \mathcal{K}(A) \cdot \frac{\|\delta(A)\|}{\|A\|} \quad \clubsuit$$

Ondorioz,  $\mathcal{K}(A) \gg 1$  handia denean, esan dezakegu sistema txarto egokituta edo txarto baldintzatua dagoela. Adibidez, Gauss algoritmoa erabiltzen bada, eta makina zenbakien mantisaren luzera  $t$  bada,  $\beta$  oinarria hartuz, froga daiteke

$$\|\delta A\|_\infty \leq n \cdot \|A\|_\infty \cdot \beta^{-t}$$

dela. Ondorioz,  $e_r \leq n \cdot \mathcal{K}(A) \cdot \beta^{-t}$  denez,  $\mathcal{K}(A) \simeq \beta^t$  edo handiagoa bada, esan dezakegu sistema txarto baldintzatua dagoela.

## 2.6 Iteraziozko Teknikak

### 2.6.1 Sarrera

- Hein haundiko matrize sakabanatuek definituriko sistemak ebazteko erabiltzen dira.
- Nuluak diren gaiak, nulu mantentzen dira normalean teknika horiekin.
- $\{\underline{x}^{(k)}\}_{k \geq 0}$  segida bat sortzen da, zenbait baldintzapetan konbergentea dena.
- **Metodo orokorra:**  $A \cdot \underline{x} = \underline{b}$  sistemaren baliokidea den  $\underline{x} = T \cdot \underline{x} + \underline{c}$  beste sistema bat definitzen da, eta azken hori erabiliz,

$$\underline{x}^{(k+1)} = T \cdot \underline{x}^{(k)} + \underline{c}$$

formula erabiltzen da bektoreen segida definitzeko.

- **Oharra:** Konbergentzia ez dago bermatuta, orokorrean.
- **Gelditze irizpidea:** Zehaztasuna ona izan arte (adibidez,  $\|\underline{x} - \underline{x}^{(k)}\| < \varepsilon$  izan arte), edo beste batzutan, iterazio kopurua muga daiteke hasieratik, metodo zuzen baten bidez baino eragiketa gutxiago egiteko.

### 2.6.2 Jacobi metodoa

$a_{ii} \neq 0$  bada  $\forall i = 1, \dots, n$ , orduan aska dezakegu ezezagun bakoitza, honako formula hauek lortuz:

$$x_i = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n a_{ij} \cdot x_j \right) \implies x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n a_{ij} \cdot x_j^{(k)} \right) \quad i = 1, \dots, n$$

(ADIBIDEA)

Adierazpen matriziala honako hau da:  $\underline{x}^{(k+1)} = T_J \cdot \underline{x}^{(k)} + \underline{c}$ , non  $T_J = (t_{ij})$  eta  $\underline{c} = (c_i)$ ,

$$t_{ij} = \begin{cases} 0 & i = j \text{ bada;} \\ -\frac{a_{ij}}{a_{ii}} & i \neq j \text{ bada.} \end{cases} \quad \text{eta} \quad c_i = \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n$$

izanik.

#### Gelditze-irizpidea

Jacobi metodoak sorturiko segidak  $\{\underline{x}^{(k)}\}_{k \geq 1}$  konbergitzen du  $\Leftrightarrow \|\underline{x} - \underline{x}^{(k)}\| \xrightarrow{k \rightarrow \infty} 0$

$\Leftrightarrow \|\underline{x}^{(k+1)} - \underline{x}^{(k)}\| \xrightarrow{k \rightarrow \infty} 0$  (azken propietate honi *Cauchy-ren propietatea* deritzo).  
Ondorioz,  $\varepsilon > 0$  finkaturik,

$$\|\underline{x}^{(k+1)} - \underline{x}^{(k)}\| < \varepsilon \quad \text{edo} \quad \frac{\|\underline{x}^{(k+1)} - \underline{x}^{(k)}\|}{\|\underline{x}^{(k+1)}\|} < \varepsilon$$

denean, geldituko dugu prozesua (normalean,  $\|\cdot\|_\infty$  norma erabiliko dugu).

### Interpretazio matrizial

$A = D - B - G$  deskonposizioa aintzakotzat hartu, non

$$D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}, \quad -B = \begin{pmatrix} 0 & \dots & \dots & 0 \\ a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{pmatrix}, \quad -G = \begin{pmatrix} 0 & a_{12} & \dots & a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & a_{n-1,n} \\ 0 & \dots & \dots & 0 \end{pmatrix}$$

diren. Orduan, badaukagu:

$$A \cdot \underline{x} = \underline{b} \Leftrightarrow D\underline{x} - B\underline{x} - G\underline{x} = \underline{b} \Leftrightarrow D\underline{x} = (B + G)\underline{x} + \underline{b} \Leftrightarrow \underline{x} = D^{-1}(B + G)\underline{x} + D^{-1}\underline{b},$$

eta iteraziozko legea

$$\underline{x}^{(k+1)} = D^{-1} \cdot (B + G)\underline{x}^{(k)} + D^{-1}\underline{b}$$

bezala idatz daiteke. Hortaz,  $T_J = D^{-1} \cdot (B + G)$  dugu.

### Oharrak

- Ez da asko erabiltzen, Gauss-Seidel metodoa normalean eraginkorragoa delako;
- $\exists i : a_{ii} = 0$  bada, orduan ekuazioak permuta ditzakegu  $a_{ii}$  gaia ahalik eta haundiena izateko asmoz (zenbaki txikietatik ez zatitzeko).
- Hasieratik normalizatzen badugu (errenkada bakoitza  $a_{ii}$ -rekin zatituz),  $n$  zatidura aurreztuko ditugu iterazio bakoitzean. Errenkada bakoitzean  $k$  gai ez-nulu badaude, eta  $k$  baino iterazio gehiago egiten baditugu, orduan merezi du normalizatzeak.

### 2.6.3 Gauss-Seidel metodoa

- Jacobi metodoaren “hobekuntza” bat da. Iterazio bakoitzean,  $\underline{x}$ -ren eguneraturiko gaiak aprobeztatuko ditugu, hots,

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) \quad i = 1, \dots, n$$

(ADIBIDEA)

- **Interpretazio matrizial:** Lehen bezala,  $A = D - B - G$  hartuz, badaukagu:

$$A \cdot \underline{x} = \underline{b} \Leftrightarrow D\underline{x} - B\underline{x} - G\underline{x} = \underline{b} \Leftrightarrow (D - B)\underline{x} = G\underline{x} + \underline{b}.$$

Eta hortaz, badaukagu

$$(D - B)\underline{x}^{(k+1)} = G\underline{x}^{(k)} + \underline{b} \Leftrightarrow \underline{x}^{(k+1)} = (D - B)^{-1} \cdot G\underline{x}^{(k)} + (D - B)^{-1}\underline{b}$$

Iteraziozko matrizea  $T_{GS} = (D - B)^{-1} \cdot G$  da.

- **Memorian biltegitzea:** Jacobi metodoan,  $n$  luzerako 2 bektore dira beharrezkoak ( $\underline{x}^{(k+1)}$  eta  $\underline{x}^{(k)}$  bektoreetarako), baina Gauss-Seidel-en metodoan,  $x_i^{(k+1)}$  kalkulaturik,  $x_i^{(k)}$  gainean idatz daiteke, azken hori beharrezkoa izango ez baita. Ondorioz,  $n$  luzerako bektore bat nahikoa izango da.

### 2.6.4 Iteraziozko metodoen konbergentzia

**2.16 teorema.**  $\underline{x}^{(0)} \in \mathbb{R}^n$  edozein bektoretarako,  $\underline{x}^{(k+1)} = T \cdot \underline{x}^{(k)} + \underline{c}$  ( $\underline{c} \neq \underline{0}$  izanik) iteraziozko metodoaz eraikitako segidak  $A \cdot \underline{x} = \underline{b}$  sistemaren soluzio zehatzerantz konbergitzen du baldin eta soilik baldin  $\rho(T) < 1$  bada.

Badaukagu  $\rho(T) = \text{Inf}\{\|T\| : \|\cdot\| \text{ norma naturala da}\}$ . Ondorioz,

**2.17 korolaria.**  $\underline{x}^{(0)} \in \mathbb{R}^n$  edozein bektoretarako,  $\underline{x}^{(k+1)} = T \cdot \underline{x}^{(k)} + \underline{c}$  ( $\underline{c} \neq \underline{0}$  izanik) iteraziozko metodoaz eraikitako segidak  $A \cdot \underline{x} = \underline{b}$  sistemaren soluzio zehatzerantz konbergitzen du baldin eta soilik baldin existitzen bada  $\|\cdot\|$  norma natural bat, zeinerako  $\|T\| < 1$  baita.

Gainera,  $\|\cdot\|$  edozein norma naturaltarako, honako bornaketa hauek dauzkagu:



- i)  $\|\underline{x} - \underline{x}^{(k)}\| \leq \|T\|^k \|\underline{x}^{(0)} - \underline{x}\|$
- ii)  $\|\underline{x} - \underline{x}^{(k)}\| \leq \frac{\|T\|^k}{1 - \|T\|} \|\underline{x}^{(1)} - \underline{x}^{(0)}\|$

**Oharrak:**

- Zenbat eta  $\|T\|$  balioa txikiago, orduan eta bizkorrago konbergituko du metodoak;
- Zenbat eta  $\rho(T)$  balioa txikiago, orduan eta bizkorrago konbergituko du metodoak.
- Badaukagu  $\|\underline{x} - \underline{x}^{(k)}\| \leq \rho(T)^k \|\underline{x}^{(0)} - \underline{x}\|$

**2.18 teorema.** *A matrizea diagonal hertsiki menperatzaileduna bada, orduan  $\underline{x}^{(0)} \in \mathbb{R}^n$  edozein bektoretarako, Jacobi eta Gauss-Seidel metodoek  $A \cdot \underline{x} = \underline{b}$  sistemaren soluzio zehatzerantz konbergitzen dute.*

*Frogapena:* Bakarrik frogatuko dugu Jacobi-ren metodorako:

$$\begin{aligned} \|T\|_{\infty} &= \max_i \sum_{j=1}^n |t_{ij}| \\ &= \max_i \sum_{j=1, j \neq i}^n \left| \frac{a_{ij}}{a_{ii}} \right| \\ &= \max_i \frac{1}{|a_{ii}|} \sum_{j=1, j \neq i}^n |a_{ij}|, \end{aligned}$$

eta  $A$  diagonal hertsiki menperatzaileduna denez,  $\sum_{j=1, j \neq i}^n |a_{ij}| < |a_{ii}|$ . Hortaz,

$$\|T\|_{\infty} < \max_i \frac{|a_{ii}|}{|a_{ii}|} = 1$$

Amaitzeko, 2.17 korolaria erabiliz, Jacobi metodoak konbergitzen duela ondorioztatzen dugu. ♣

(ADIBIDEA)

**2.19 teorema.** *A matrizea simetrikoa eta positiboki definitua bada, orduan  $\underline{x}^{(0)} \in \mathbb{R}^n$  edozein bektoretarako, Gauss-Seidel metodoak  $A \cdot \underline{x} = \underline{b}$  sistemaren soluzio zehatzerantz konbergitzen du.*

(ADIBIDEA)

**2.20 teorema** (Stein-Rosenberg).  $a_{ij} \leq 0 \quad \forall i \neq j$  eta  $a_{ii} > 0 \quad \forall i = 1, \dots, n$  betetzen badira, orduan honako propietate hauen artean, bat eta soilik bat egiaztatzen da:

$$i) \quad 0 < \rho(T_{GS}) < \rho(T_J) < 1;$$

$$ii) \quad 1 < \rho(T_J) < \rho(T_{GS});$$

$$iii) \quad \rho(T_J) = \rho(T_{GS}) = 1;$$

$$iv) \quad \rho(T_J) = \rho(T_{GS}) = 0,$$

non  $T_J$  Jacobi metodoari dagokion iteraziozko matrizea den, eta  $T_{GS}$  Gauss-Seidel metodoari dagokiona.

Azken egoera hori tipikoa da ekuazio diferentzialen esparruan. Gertatzen denean, teoremak dioenez, Jacobi metodoak kongergitzen duenean, Gauss-Seidel metodoak ere kongergitzen du, bizkorrago.

(ADIBIDEA)

### 2.6.5 Lasaikuntza Teknikak

Gauss-Seidel metodoko ekuazioak berridazten baditugu  $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \dots$  eran, ondorengoa lortzen dugu:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^n a_{ij} x_j^{(k)} \right) \quad i = 1, \dots, n$$

bigarren batugaia interpretatu daiteke  $x_i^{(k+1)}$  eta  $x_i^{(k)}$  gaien arteko distantziatzat. Zuzenketa hori alda dezakegu  $\omega > 0$  parametro batekin biderkatuz, honako hau lortuz:

$$x_i^{(k+1)} = x_i^{(k)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^n a_{ij} x_j^{(k)} \right) \quad i = 1, \dots, n$$

eta hortaz, erabiliko ditugun ekuazioak, lasaikuntza metodoari dagozkionak:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij} x_j^{(k)} \right) \quad i = 1, \dots, n.$$

- $\omega = 1 \implies$  Gauss-Seidel.
- $\omega > 1 \implies$  Lasaikuntzagaineko metodoa (S.O.R. = Sucessive Over Relaxation). Erabili ohi da Gauss-Seidelek monotonoki konbergente denean, konbergentzia bizkortzeko;
- $\omega < 1 \implies$  Azpilasaikuntza metodoa. Erabili ohi da Gauss-Seidelek konbergitzen ez duenean, edo oszilatzen duenean.
- **Interpretazio matritziala** Lasaikuntza metodoaren ekuazioen bi atalak  $a_{ii}$ -rekin biderkatuz:

$$a_{ii}x_i^{(k+1)} + \omega \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} = a_{ii}(1-\omega)x_i^{(k)} + \omega b_i - \sum_{j=i+1}^n \omega a_{ij}x_j^{(k)}$$

Hortaz,  $A = D - B - G$  berriro hartuz,

$$(D - \omega B)\underline{x}^{(k+1)} = \omega \underline{b} + [(1 - \omega)D + \omega G]\underline{x}^{(k)}$$

eta iteraziozko ekuazioak:

$$\underline{x}^{(k+1)} = \omega(D - \omega B)^{-1}\underline{b} + (D - \omega B)^{-1} \cdot [(1 - \omega)D + \omega G]\underline{x}^{(k)}$$

Iteraziozko matrizea  $T_\omega = (D - \omega B)^{-1} \cdot [(1 - \omega)D + \omega G]$  da.

**2.21 teorema** (Kahan).  $a_{ii} \neq 0 \quad \forall i = 1, \dots, n$  bada, orduan  $\rho(T_\omega) \geq |\omega - 1|$ .

Hortaz, lasaikuntza teknika konbergentea izateko,  $0 < \omega < 2$  izan behar du, derrigorrez.

**2.22 teorema** (Ostrowski-Reich).  $A$  matrizea simetrikoa, positiboki definitua bada eta  $0 < \omega < 2$  bada, orduan lasaikuntza metodoak konbergitzen du edozein  $\underline{x}^{(0)} \in \mathbb{R}^n$  bektoretarako.

**2.23 teorema.**  $A$  matrizea simetrikoa, positiboki definitua bada eta gainera tridigonal bada, orduan:

- *i)*  $\rho(T_{GS}) = \rho(T_J)^2 < 1$ ;
- *ii)* lasaikuntza metodorako  $\omega$  optimoa lor daiteke, eta honako hau da:

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(T_J)^2}} = \frac{2}{1 + \sqrt{1 - \rho(T_{GS})}}$$

Azken teorema honen baldintzetan,  $\rho(T_\omega) = \omega - 1$  da.

(ADIBIDEA)



---

# 3. kapitulua

---

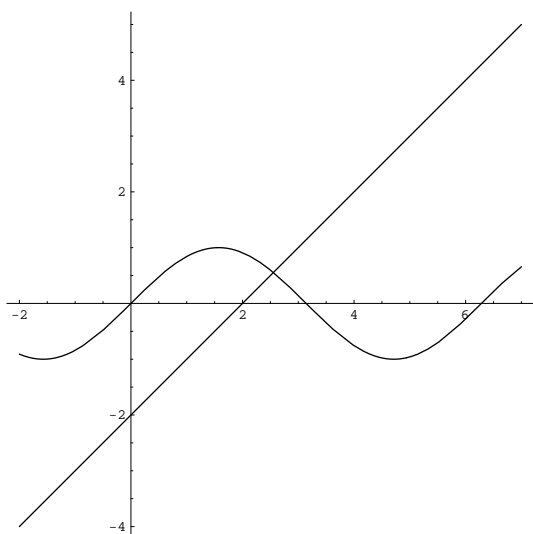
## Ekuaizio ez linealen ebazpen hurbildua

### 3.1 Sarrera

- $f(x) = 0$  erako ekuazioak ebazteko, hainbat iteraziozko metodo ikusiko ditugu.
- **Erroen banaketa:** Metodoak aplikatu baino lehen, erro bakoitza daukan  $[a, b]$  tarte bat aurkitzen da. Horretarako, bi teknika erabiltzen dira:
  - i) *Bolzanoren Teorema:*  $f: [a, b] \rightarrow \mathbb{R}$  jarraikia bada eta  $f$ -ren zeinua aldatzen bada tartean (i.e.,  $f(a) \cdot f(b) < 0$  bada), orduan existitzen da  $p \in [a, b]$ ,  $f(p) = 0$  betetzen duena.
  - ii) *Adierazpen grafikoaren erabilera:*  $f(x) = f_1(x) - f_2(x)$  deskonposizioa emanda,  $f_1$  eta  $f_2$  funtzioen grafikoak marrazten dira. Orduan,  $p$  ebakidura abszisa bakoitza erro bat izango da,  $f_1(p) = f_2(p) \Rightarrow f(p) = f_1(p) - f_2(p) = 0$  delako.

Normalean, biak erabiltzen dira batera, ondorengo adibidean ikusten denez:

**3.1 adibidea.**  $f(x) = \sin x - x + 2$  funtzioaren erroak kokatuko ditugu:



$f(x) = \sin x - (x - 2)$  dugu. Begibistaz,  $f_1(x) = \sin x$  eta  $f_2(x) = x - 2$  funtzioen grafikoak  $(2, \pi)$  tarteko puntu batean ebakitzen dira, eta inon ere ez. Egiazta dezakegu azken hori zehaztasunez:

$$f(2) = \sin 2 > 0 \quad \text{eta} \quad f(\pi) = -\pi + 2 < 0$$

denez, Bolzanoren teoremaz,  $\exists c \in [2, \pi]$ , non  $f(c) = 0$  egiaztatzen den.

Bestalde,  $f_2$  gorakorra denez, ondoriozta daiteke erraz ebakidura puntu gehiagorik ez daudela.

- Hurrengo taulan, metodo motak konparatzen dira:

Metodo Mota	Konbergentzia	Abiadura	Zeren bidez?
ITZIAK	Konbergenteak, beti	Astiro konbergitzen dute	tarte familia beherakor bat
IREKIAK	Ez dago bermaturik	itxiak baino azkarragoak	$\{x_n\}_{n \geq 0} \rightarrow p$

Normalean, erroak tarte finitutan kokaturik, lehen metodo itxi bat erabiltzen da tartearen luzera murrizteko. Gero, konbergentzia bermaturik, aplikatzen da metodo ireki bat erroa kalkulatzeko.

## 3.2 Metodo Itxiak

### 3.2.1 Bisekzio metodoa (Bolzano metodoa)

- Hasierako tarteak  $[a, b]$  izanik, definitzen dira:

$$a_0 = a, \quad \text{eta} \quad b_0 = b.$$

Hemendik aurrera,  $n$ -garren etapan,  $[a_n, b_n]$  tartearen erdigunea kalkulatzeko da:

$$x_n = \frac{a_n + b_n}{2}$$

eta ondorengo tartea kalkulatzeko erabiltzen da azpiko irizpide hau, Bolzanoren teoreman oinarrituta:

$$\begin{cases} f(x_n) \cdot f(a_n) < 0 & \text{bada, } \Rightarrow \begin{cases} a_{n+1} = a_n \\ b_{n+1} = x_n \end{cases} \\ f(x_n) \cdot f(b_n) < 0 & \text{bada, } \Rightarrow \begin{cases} a_{n+1} = x_n \\ b_{n+1} = b_n \end{cases} \end{cases}$$

- **Gelditze-irizpidea:** Pausu bakoitzean,  $x_n$  kalkulatu ondoren,  $f(x_n) = 0$  bada (txiripaz), prozesua gelditzen da. Bestalde,  $\varepsilon > 0$  aldezturik,  $|f(x_n)| < \varepsilon$  bada, prozesua gelditzen da. Baina irizpiderik erabilena tartearen luzera da.
- **Konbergentzia:** Beti konbergitzen du hasieran Bolzanoren Teoremaren baldintzak betetzen badira:  $p$  soluzio zehatza bada, lehen pausuan

$$|x_0 - p| \leq \frac{|b_0 - a_0|}{2} = \frac{b - a}{2}$$

Ondorengo pausuan,

$$|x_1 - p| \leq \frac{|b_1 - a_1|}{2} = \frac{|b_0 - a_0|}{2 \cdot 2} = \frac{b - a}{2^2}$$

eta, orokorrean,  $n$ -garren pausuan,

$$|x_{n-1} - p| \leq \frac{|b_{n-1} - a_{n-1}|}{2} = \dots = \frac{b - a}{2^n} \xrightarrow{n \rightarrow \infty} 0.$$

Azken formula erabiliz, kalkula dezakegu behar diren pausu kopurua zehatzasun jakin bat lortzeko.  $\varepsilon > 0$  finkaturik,

$$|x_n - p| \leq \frac{b - a}{2^{n+1}} \leq \varepsilon \iff 2^{n+1} \geq \frac{b - a}{\varepsilon},$$

eta hemendik abiatuz, logaritmoak hartuz (eta logaritmoa gorakorra izanik),

$$(n + 1) \log 2 \geq \log(b - a) - \log \varepsilon \iff n \geq \frac{\log(b - a) - \log \varepsilon}{\log 2} - 1.$$

**3.2 adibidea.** Bolzano Metodoaz,  $x^3 - 2 = 0$  erro errealak kalkulatu nahi dugu  $[1, 2]$  tartearekin hasiz. Zenbat iterazio egin behar dira  $\varepsilon = 10^{-6}$  zehatzasuna lortzeko?

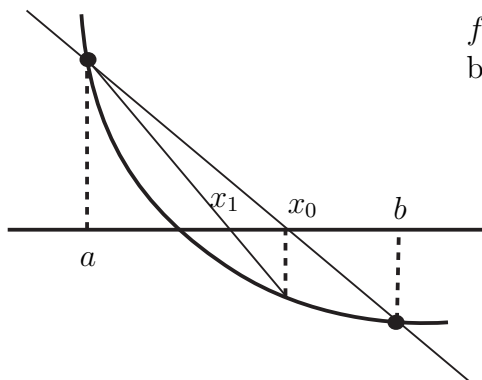
$$n \geq \frac{\log(2 - 1) - \log 10^{-6}}{\log 2} - 1 = \frac{6 \log 10}{\log 2} - 1 \simeq 18.93 \Rightarrow n = 19.$$

Ondorioz, 19garren iterazioarekin,  $\sqrt[3]{2}$  lortu egingo dugu  $10^{-6}$  baino txikiagoa den errorez.

- **Oharrak:**

- Bakarrik erabili dugu  $[a, b]$  tartea eta ez  $f(x) = x^3 - 2$  dela.
- Goiko estimazioa egindako errorearen goi-borne bat da, besterik ez.
- Baliteke  $x_n$   $x_{n+1}$  baino hurbilpen hobeagoa izatea (Kontuz!)
- Nahiko metodo motela da. Gainera, funtzio bat ebaluatzea prozesu garestia da. Dena dela, metodo ireki bat erabili aurretik egokia gertatzen da.
- Metodoak funtzionatzeko, nahitaezkoa da  $f(a) \cdot f(b) < 0$  izatea;

### 3.2.2 Lekugune faltsuaren metodoa (*Regula falsi*)



$f(a) \cdot f(b) < 0$  delako kasuan,  $[a_n, b_n]$  familia beherakor bat definituko dugu honela:

- Pausu bakoitzean,  $(a_n, f(a_n))$  eta  $(b_n, f(b_n))$  puntuetatik igarotzen den lerro zuzena hartzen da aintzartat. Zuzen hau eta  $y = 0$  ardatz horizontalearekiko ebakitze-puntuari  $x_n$  deitzen zaio.
- $a_{n+1}$  edo  $b_{n+1}$  definitzeko, Bolzanoren metodoan erabiliko irizpidea erabiltzen da.

- **Adierazpen analitikoa:**  $n$ -garren pausuko zuzenaren ekuazioa:

$$\frac{y - f(b_n)}{f(b_n) - f(a_n)} = \frac{x - b_n}{b_n - a_n}.$$

Hortaz,

$$y - f(b_n) = \frac{f(b_n) - f(a_n)}{b_n - a_n}(x - b_n).$$

Orain,  $y = 0$  eginez, ondoriozta dezakegu  $x_n$  ebakidura puntuaren abszisa:

$$x_n = b_n - \frac{b_n - a_n}{f(b_n) - f(a_n)} f(b_n) = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}$$

**Oharra:**  $f(a_n) \cdot f(b_n) < 0$  denez, orduan  $f(b_n) \neq f(a_n)$ , eta ondorioz, azken izendatzailea ez da nulua.

- **Konbergentzia**

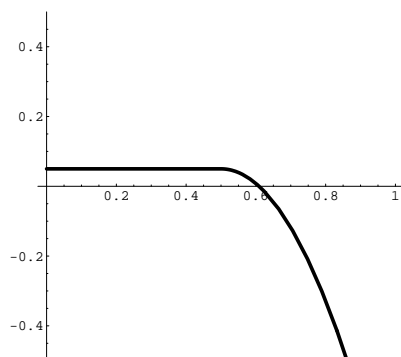


- i) Froga daiteke metodo hau konbergentea dela, zenbait baldintza betetzen badira:

$$\left. \begin{array}{l} f(x) \text{ jarraikia bada, eta} \\ \exists p \in [a, b] : f(p) = 0 \end{array} \right\} \implies x_n \xrightarrow{n \rightarrow \infty} p.$$

- ii) Orokorrean, bisekzio metodoa baino arinagoa da (adibidez,  $x^3 - 2 = 0$ ,  $[1, 2]$  tartean), baina hori ez da beti gertatzen, honako adibide honetan ikusten denez:

### 3.3 adibidea.



Kontsideratu

$$f(x) = \begin{cases} \delta & x \in [0, 1/2] \\ 4(1 + \delta)(x - x^2) - 1 & x \in [1/2, 1] \end{cases}$$

funtzioa. Ikusiriko bi metodo itxiak aplikatzen baditugu  $f$ -k  $[0, 1]$  tartean duen erroa kalkulatzeko, konproba daiteke  $\delta$  txikia bada, bisekzio metodoak lekugune faltsuak baino hobeto funtzionatzen duela. Irudia egiteko,  $\delta = 0.05$  hartu dugu.

- **Gelditze-irizpidea:** Hainbat kasutan, hasierako tarteko mutur bat finko gertatzen da iterazio bakoitzean (azken adibidean,  $b = b_1 = b_2 = \dots$ ). Horregaitik, beste muturrak errorantz konbergitzen du (azken adibidean,  $a_n$  segidak). Hortaz,  $|b_n - a_n|$  ez du joko zerorantz, eta  $|b_n - a_n| < \varepsilon$  gelditze irizpideak ez dauka zentzurik (Bisekzio metodoan, ordea, bai).

Bestalde, badakigu  $x_n$  segidak ebazpen zehatzerantz konbergitzen duela. Horregaitik, honako gelditze-irizpide hauek erabil daitezke:

$$|x_n - x_{n-1}| < \varepsilon \quad \text{edo} \quad |f(x_n)| < \varepsilon.$$

Funtzioaren ebaluaketa saihesteko (garestia da!), lehena erabi ohi da. Bertsio erlatiboan ere erabil daiteke.

## 3.3 Metodo Irekiak

### 3.3.1 Puntu Finkoko Iterazioa

**3.1 definizioa.** Esango dugu  $x \in \mathbb{R}$   $g$  funtzioaren puntu finko bat dela baldin  $g(x) = x$  bada.

Metodo honetan,  $f(x) = 0$  ekuazioa ebatzi beharrea  $g(x) = x$  ekuazio baliokide bat ebatziko da. Hots,  $g$  beste funtzio baten puntu finkoak aurkituko ditugu. Adibidez,  $g(x) = x - f(x)$  funtzioaren  $p$  puntu finko bat aurkitzen badugu, orduan

$$p = g(p) = p - f(p) \Rightarrow f(p) = 0.$$

Gehienetan,  $x - f(x)$  ez da funtziorik egokiena erabiltzeko.

**3.1 teorema** (Puntu finkoko teorema). *Demagun  $g: [a, b] \rightarrow [a, b]$  jarraitua dela. Orduan,*

$$i) \exists p \in [a, b] : g(p) = p;$$

ii)  *$g$  deribagarria bada  $[a, b]$  tartean, eta honako propietate hau betetzen bada:*

$$\exists K < 1 : |g'(x)| \leq K \quad \forall x \in [a, b], \quad (3.1)$$

*orduan  $p$   $g$ -ren puntu finko bakarra da  $[a, b]$  tartean.*

iii) *Baldin honako propietate hau betetzen bada:*

$$\exists K < 1 : \forall x_1, x_2 \in [a, b], |g(x_2) - g(x_1)| \leq K|x_2 - x_1|, \quad (3.2)$$

*orduan  $p$   $g$ -ren puntu finko bakarra da  $[a, b]$  tartean.*

**Oharra:** (3.2) propietateari Lipschitz baldintza deritzo, eta  $K$  konstanteari, Lipschitz konstantea.

*Frogapena:*

Lehen, *i) frogatuko dugu.*  $g(a) = a$  edo  $g(b) = b$  bada, amaitu egin dugu. Orduan, suposa dezakegu  $g(a) > a$  eta  $g(b) < b$  direla.  $h(x) = g(x) - x$  funtzioa aintzakotzat hartuz,

$$\left. \begin{array}{l} h(a) = g(a) - a > 0 \\ h(b) = g(b) - b < 0 \end{array} \right\} \stackrel{\text{Bolzano}}{\Rightarrow} \exists p \in [a, b] : h(p) = 0 \iff g(p) = p.$$

Bigarrenez, ikusiko dugu (3.1) propietatea betetzen bada, orduan (3.2) propietatea (Lipschitz baldintza) betetzen dela. Batazbesteko Balioaren Teoremaz, badaukagu

$$\exists \xi \in [a, b] : g(x_2) - g(x_1) = g'(\xi)(x_2 - x_1)$$

Eta orain, balio absolutuak hartuz eta (3.1) propietatea erabiliz, badaukagu

$$|g(x_2) - g(x_1)| \leq K|x_2 - x_1|,$$

hau da,  $g$ -k Lipschitz baldintza betetzen du.

Amaitzeko, bakarrik geratzen zaigu  $g$ -k Lipschitz baldintza betetzen duenean, puntu finko bakar bat duela frogatzea. Horretarako, suposa dezagun  $p, q \in [a, b]$   $g$ -ren puntu finkoak direla. Orduan,

$$|p - q| = |g(p) - g(q)| \leq K|p - q| \Rightarrow |p - q| \leq K|p - q|$$

Orain,  $p \neq q$  izango balitz, orduan  $K \geq 1$  izango litzateke (kontraesana). Orduan, derrigorrez,  $p = q$  dugu. ♣

Orain deskribatuko dugu puntu finkoko teknika  $f(x) = 0$  ekuazioa ebazteko:

1. Bilatu  $g$  funtzio jarraitu bat, honako hau betetzen duena:

$$g(x) = x \text{ bada, } \Rightarrow f(x) = 0;$$

2. Hartu  $x_0$  hastapen hurbilpena, eta definitu  $\{x_n\}_{n \geq 0}$  segida azpiko formula honetaz:  $x_{n+1} = g(x_n) \quad \forall n \geq 0$ .
3.  $\{x_n\}_{n \geq 0}$  segida konbergentea bada eta  $p$  bere limitea bada, orduan,  $g$  jarraitua denez, ondoriozta dezakegu  $g(p) = p$ . Beraz,

$$g(p) = g\left(\lim_{n \rightarrow \infty} x_n\right) = \lim_{n \rightarrow \infty} g(x_n) = \lim_{n \rightarrow \infty} x_{n+1} = p.$$

4. **Gelditze-irizpidea:** beste metodoetan bezala,  $|x_n - x_{n-1}| \leq \varepsilon$  izan arte (edo bertsio erlatiboa erabiliz).

**3.2 teorema** (Konbergentzia globaleko teorema). *Bedi  $g: [a, b] \rightarrow [a, b]$  funtzio jarraitu bat, Lipschitz baldintza betetzen duena  $K < 1$  konstantez. Orduan,  $x_{n+1} = g(x_n)$  formulaz eraikitako segidak  $g$ -ren puntu finko bakarrentz jotzen du edozein  $x_0 \in [a, b]$  hastapen baliotarako.*

*Frogapena:*  $g$  Lipschitz izateagatik, badakigu  $p$  puntu finko bakar bat duela. Orduan, Lipschitz izateagatik,

$$|x_n - p| = |g(x_{n-1}) - g(p)| \leq K|x_{n-1} - p|$$

Azken formula behin eta berriro aplikatuz:

$$|x_n - p| \leq K|x_{n-1} - p| \leq K^2|x_{n-2} - p| \leq \dots \leq K^n|x_0 - p|.$$

Orain, limiteak hartuz,

$$\lim_{n \rightarrow \infty} |x_n - p| \leq \lim_{n \rightarrow \infty} K^n|x_0 - p| = 0,$$

$K < 1$  delako. Frogatu egin dugu  $x_n \xrightarrow{n \rightarrow \infty} p$  jotzen duela. ♣

**Oharrak:**

- i)  $g$ -k (3.1) baldintza betetzen badu, orduan Lipschitz propietatea beteko du, eta ondorioz, teorema egiazkoa izango da, ere.
- ii)  $g$  funtzioaren hautaketa oso garrantzitsua da (ADIBIDEA).

**3.3 teorema.** (3.2) *Teoremaren baldintzetan, badaukagu:*

$$|x_n - p| \leq \frac{K^n}{1 - K} |x_1 - x_0|,$$

$K$  Lipschitz konstantea delarik.

**Oharrak:**

- Zenbat eta  $K$  txikiago, orduan eta arinago joko du metodoak ebazpen zehatzerantz.
- $[a, b]$  tartea txikitzen bada,  $K$  alda daiteke.

**3.4 teorema** (Konbergentzia lokaleko teorema). *Bedi  $g: [a, b] \rightarrow \mathbb{R}$  funtzio bat, honako propietate hauek betetzen dituena:*

- i)  $\exists p \in [a, b] : g(p) = p;$
- ii)  $g$ -ren lehen deribatua jarraitua da;
- iii)  $|g'(p)| < 1.$

Orduan,  $\exists \varepsilon > 0$  zeinerako  $|x_0 - p| < \varepsilon$  bada, puntu finkoko iterazioak sorturiko segidak  $p$ -rantz jotzen du.

*Frogapena:*

IDEIA:  $|g'(p)| < 1$  eta  $g'$  jarraitua denez,  $|g'(x)| < 1$  mantenduko da  $p$ -ren ingurune batean, eta hor, funtzioa Lipschitz izanik, (3.2) Teorema erabiliz, metodoa konbergentea dela ondoriozta dezakegu.

Lehen, ikusiko dugu  $g$  Lipschitz dela:  $|g'(p)| < 1$  denez,  $\exists K \in \mathbb{R}$ , zeinerako  $|g'(p)| < K < 1$  den. Orduan,  $g'$  jarraitua izanik,  $\exists \varepsilon > 0$  zeinerako

$$|p - x| < \varepsilon \implies |g'(x)| < K$$

dugu. Frogatu egin dugu  $g$  Lipschitz dela  $(x - p, x + p)$  tartean.

Bakarrik geratzen zaigu

$$g: [p - \varepsilon, p + \varepsilon] \longrightarrow [p - \varepsilon, p + \varepsilon]$$

doala frogatzea, hots,  $g((p - \varepsilon, p + \varepsilon)) \subseteq (p - \varepsilon, p + \varepsilon)$  dela. Horretarako, har dezagun  $x \in (p - \varepsilon, p + \varepsilon)$ . Frogatu behar dugu  $g(x) \in (p - \varepsilon, p + \varepsilon)$ . Batazbesteko Balioaren Teoremaz,

$$\exists \xi \in (x, p) : g(x) - g(p) = g'(\xi)(x - p),$$

eta hortaz, balio absolutuak hartuz,

$$|g(x) - p| = |g(x) - g(p)| = |g'(\xi)| \cdot |x - p| < K|x - p| < K\varepsilon < \varepsilon,$$

frogatu nahi genuen lez. ♣

**Oharra:** Azken teorema hori aplikatzerakoan, ez dugu  $p$  puntu finkoa ezagutuko. Estrategia:  $p$  tarte txiki batean kokaturik, ebalua dezakegu  $|g'(x)|$  tartearen bi muturretan  $|g'(p)|$  balioaren hurbilpenak izateko.  $g'$  jarraitua bada, orduan hurbilpen horiek fidagarriak izan daitezke eta (3.4) teorema aplika dezakegu.

(ADIBIDEA)

## 3.4 Iteraziozko metodoen konbergentzia

Kontsideratu  $f(x) = 0$  ekuazioaren  $p$  erro bat. Bedi  $\{x_n\}_{n \geq 0} \xrightarrow{k \rightarrow \infty} p$  zenbait iteraziozko metodok sorturiko segida. Orduan,  $n$ -garren pausuan egindako erroa  $e_n = x_n - p$  dugu.

**3.2 definizioa.**  $\{x_n\}_{n \geq 0}$  segidaren  $p$ -rantzeko konbergentzia lehen ordenakoa dela esango dugu baldin eta

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|} = C$$

bada,  $C \in (0, 1)$  izanik.

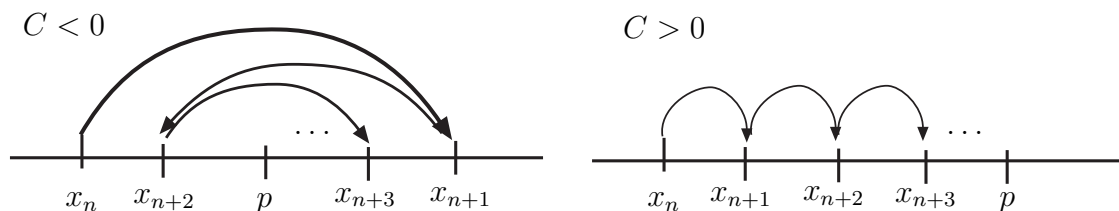
**Oharra:** Lehen ordenako konbergentzia badugu, orduan, asintotikoki,  $|e_{n+1}| \simeq C|e_n|$  egiaztatzen da.

**3.3 definizioa.**  $\{x_n\}_{n \geq 0}$  segidaren  $p$ -rantzeko konbergentzia lehen ordena linealekoa dela esango dugu baldin eta

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = C$$

badugu,  $C \in (-1, 1)$  eta  $C \neq 0$  izanik.

**Oharra:** Kasu honetan, segida konbergentea denez,  $n_0$  jakin batetik aurrera  $e_{n+1}/e_n$  zatidura guztiak zeinu berekoak izango dira. Ondorioz,  $C$ -ren zeinuaren arabera, konbergentzia monotono ala oszilagarria izango dugu.



**3.4 definizioa.**  $\{x_n\}_{n \geq 0}$  segidaren  $p$ -rantzeko konbergentzia  $\alpha \in (1, \infty)$  ordenakoa dela esango dugu baldin eta

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^\alpha} = K > 0 \text{ bada,}$$

hots, asintotikoki,  $|e_{n+1}| \simeq K|e_n|^\alpha$ , non  $K > 0$ .

**3.5 definizioa.** Esango dugu metodo batek *koadratikoki* konbergitzen duela baldin eta  $\alpha = 2$  ordenako konbergentziaz konbergitzen badu.  $\alpha = 3$  ordenako konbergentziaz konbergitezotan, esango dugu metodoaren konbergentzia *kubikoa* dela.

**Oharrak:**

- Azken definizio hori balio absoluturik gabe enuntzia daiteke.
- Nabaritu  $\alpha$  haundia edo  $K$  txikia den kasuan, konbergentzia arina izango dela, abiaduran  $\alpha$ -ren eragina inportanteagoa izanik.

### 3.4.1 Puntu finko metodoen konbergentzia

Demagun  $f(x) = 0$  ekuazioaren  $p$  soluziorantz konbergitzen duen puntu finkoko iterazioa erabili dugula, iteraziozko funtzioa  $g$  izanik. Suposa dezagun, orokortasuna galdu gabe,  $x < p$  dela. Orduan, Taylor garapena erabiliz,  $x$   $p$ -ren hurbil baldin badago,

$$g(x) = g(p) + g'(p)(x-p) + \frac{g''(p)}{2!}(x-p)^2 + \dots + \frac{g^{(k)}(p)}{k!}(x-p)^k + \frac{g^{(k+1)}(\xi)}{(k+1)!}(x-p)^{k+1}$$

dugu, non  $\xi \in (x, p)$  den.

$k = 0$  eta  $x = x_n$  hartuz, garapen hori Bataztesteko Balioaren Teorema da, eta hortaz,  $\exists \xi_n \in (x_{n+1}, p)$  non

$$e_{n+1} = x_{n+1} - p = g(x_n) - p = g(x_n) - g(p) = g'(\xi_n)(x_n - p) = g'(\xi_n)e_n.$$

Limiteak hartuz,  $\xi_n \xrightarrow{k \rightarrow \infty} p$  denez eta  $g'$  jarraitua denez,

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = \lim_{n \rightarrow \infty} g'(\xi_n) = g'(\lim_{n \rightarrow \infty} \xi_n) = g'(p).$$

Ondorioz,  $g'(p) \neq 0$  bada, orduan metodoaren konbergentzia lehen ordena linealekoa da.

Bestalde,  $g'(p) = 0$  bada, orduan jarrai dezakegu, eta Taylor garapena  $x = x_n$  eta  $k = 1$  erabiliz,

$$e_{n+1} = x_{n+1} - p = g(x_n) - g(p) = \overbrace{g'(p)}^{=0} (x_n - p) + \frac{g''(\xi_n)}{2} (x_n - p)^2 = \frac{g''(\xi_n)}{2} e_n^2$$

Hortaz,  $g''$  jarraitua bada,

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^2} = \lim_{n \rightarrow \infty} \frac{g''(\xi_n)}{2} = \frac{g''(p)}{2}.$$

Ondorioz,  $g''(p) \neq 0$  bada, orduan metodoaren konbergentzia koadratikoa da.

Bestalde,  $g''(p) = 0$  bada, orduan jarraitu dezakegu, eta Taylor seriea  $k = 2$  eta  $x = x_n$  kontsideratuz,

$$\begin{aligned} e_{n+1} &= x_{n+1} - p = g(x_n) - g(p) \\ &= \overbrace{g'(p)}^{=0} (x_n - p) + \overbrace{\frac{g''(p)}{2!}}^{=0} (x_n - p)^2 + \frac{g'''(\xi_n)}{3!} (x_n - p)^3 \\ &= \frac{g'''(\xi_n)}{6} e_n^3. \end{aligned}$$

Eta era berean arrazoituz jarrai dezakegu. Orokorrean,

$$g'(p) = g''(p) = \dots = g^{(m-1)}(p) = 0 \quad \text{eta} \quad g^{(m)}(p) \neq 0$$

betetzen bada, orduan metodoaren konbergentzia  $m$  ordenakoa da.

### 3.4.2 Bisekzio eta *Regula Falsi* metodoen konbergentzia

Bi metodo horietan, ezin dugu bermatu

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|} = K > 0$$

betetzen dela, ezta  $|\cdot|$  gabe ere, baina aurki ditzakegu honako bornaketa hauek:

$$|e_n| \leq C_n,$$

non

$$\lim_{n \rightarrow \infty} \frac{C_{n+1}}{C_n^\alpha} = K > 0$$

betetzen den. Hori betetzen denean, esango dugu metodoaren konbergentzia  $\alpha$  ordenakoa dela esango dugu ( $\alpha \in (1, \infty)$  delarik).

Bisekzio metodoan,  $|e_n| = |x_n - p| \leq \frac{b-a}{2^{n+1}} = C_n$  hartuz, egiaztatzen da

$$\lim_{n \rightarrow \infty} \frac{C_{n+1}}{C_n} = \frac{1}{2},$$

hots, konbergentzia linealekoa da,  $1/2$  konstantez. Beste hitzetan, errorearen estimazioa pausu bakoitzean erdibiten da.

Bestalde, lekugune faltsuaren metodoaren konbergentzia lehen ordena linealekoa dela froga daiteke (adierazitako zentzuan).

## 3.5 Konbergentzia Arintzeko Teknikak

### 3.5.1 Aitken $\Delta^2$ algoritmoa

Linealki konbergitzen duen edozein segidaren konbergentzia arintzeko erabili ohi da. (Edozein segida, ez bakarrik iteraziozko metodo batek sortuak).

Demagun  $\{x_n\}_{n \geq 0} \xrightarrow{k \rightarrow \infty} p$  linealki (i.e. lehen ordena linealeko konbergentziaz) konbergitzen duela, hots,  $\exists \lambda \in (-1, 1)$  ez-nulua zeinerako

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = \lambda \neq 0 \text{ baita.}$$

Hots,  $n$  handia denean, badaukagu:

$$e_{n+1} \approx \lambda e_n, \quad \text{eta} \quad e_{n+2} \approx \lambda e_{n+1}.$$

Hortaz,  $\lambda$  bakanduz,

$$\lambda \approx \frac{x_{n+1} - p}{x_n - p} \approx \frac{x_{n+2} - p}{x_{n+1} - p} \implies (x_{n+1} - p)^2 \approx (x_n - p)(x_{n+2} - p)$$

Azken formula garatuz, eta hortik  $p$  askatuz,

$$p \approx \frac{x_{n+2}x_n - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n} \approx x_n - \frac{A}{x_{n+2} - 2x_{n+1} + x_n}. \quad (3.3)$$

Orain, lortuko dugu  $A$ :

$$x_n \cdot (x_{n+2} - 2x_{n+1} + x_n) - A = x_{n+2}x_n - x_{n+1}^2$$



$$\implies A = x_{n+1}^2 - 2x_{n+1}x_n + x_n^2 = (x_{n+1} - x_n)^2 \quad (3.4)$$

Orain, (3.3) eta (3.4) formuletatik lortzen dugu:

$$p \approx x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}.$$

Azken formula hori erabiliko dugu  $\{y_n\}_{n \geq 0}$  segida berri bat definitzeko honela:

$$\boxed{y_n = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n} \quad \forall n \geq 0.} \quad (3.5)$$

Aurreko baldintzetan, froga daiteke  $\{y_n\}_{n \geq 0}$  segidak  $\{x_n\}_{n \geq 0}$  segidak baino bizkorrago konbergitzen duela  $p$ -rantz. Zehazkiago:

$$\lim_{n \rightarrow \infty} \frac{y_n - p}{x_n - p} = 0$$

Hala eta guztiz ere, Aitken-en  $\Delta^2$  algoritmoaz, konbergentzia koadratikoa EZ dago bermatuta.

**Kontuz!** Aurreko laukiko formula dagoen moduan erabili behar da. Zatiki bakar batez idaztekotan,

$$p \approx \frac{x_{n+2}x_n - x_{n+1}^2}{x_{n+2} - 2x_{n+1} + x_n}$$

desegonkorra izango litzateke 0/0 motako balioak direla eta.

(ADIBIDEA)

### 3.5.2 Steffensen algoritmoa

Aitken-en algoritmo “hobetua” da. Demagun puntu finkoko iterazioaz lorturiko segida linealki konbergentea dela. Orduan, Steffensen-en metodoa aplikatuz, konbergentzia koadratikoa lor dezakegu (beraz, konbergentzia linealeko baldintza ez da guztiz beharrezkoa).

ALGORITMOA:

$g$  iteraziozko funtzioa eta  $x_0$  hastapen balioa emanda, ondorengo erako segida definituko dugu:

$$x_0^{(0)}, x_1^{(0)}, x_2^{(0)}, x_0^{(1)}, x_1^{(1)}, x_2^{(1)}, \dots, x_0^{(i)}, x_1^{(i)}, x_2^{(i)}, \dots \quad (3.6)$$

ondorengo metodoaz definituta:

i) Hartu  $x_0^{(0)} = x_0$ ;

ii)  $x_0^{(i)}$  emanda, erabili puntu finkoko iterazioa ondorengo bi gaiak definitzeko, hots:

$$x_1^{(i)} = g(x_0^{(i)}) \quad \text{eta} \quad x_2^{(i)} = g(x_1^{(i)})$$

iii)  $x_0^{(i)}, x_1^{(i)}$  eta  $x_2^{(i)}$  kalkulaturik, erabili Aitken-en formula (3.5)  $x_0^{(i+1)}$  definitzeko, i.e.:

$$x_0^{(i+1)} = x_0^{(i)} - \frac{(x_1^{(i)} - x_0^{(i)})^2}{x_2^{(i)} - 2x_1^{(i)} + x_0^{(i)}}.$$

iv) burutu (ii) – (iii) behin eta berriz.

### Oharrak:

- Froga daiteke Steffensen-en metodoa benetazko iteraziozko metodo bat dela (zehazkiago,  $x_0^{(i+1)} = \psi(x_0^{(i)})$  dela,  $\psi(x) = x - \frac{(g(x)-x)^2}{g(g(x))-2g(x)+x}$  izanik.)
- **Gelditze-irizpidea:**  $|x_0^{(n)} - p| < \epsilon$  motako irizpidea erabiliko dugu (edo errore erlabiboaz).  $x_2^{(n)} - 2x_1^{(n)} + x_0^{(n)} = 0$  bada, orduan ezin izango dugu zatitu  $x_0^{(n+1)}$  kalkulatzeko, eta ondorioz  $x_2^{(n)}$  hartuko dugu metodoaren irteeratzat.  $|g'(p)| < 1$  den kasuan, (3.6) segidako ondoz ondoko gaien artean kalkula daiteke errorea.

**3.5 teorema.** Demagun  $p \in \mathbb{R}$   $g(x) = x$  ekuazioaren erro bat dela,  $g'(p) \neq 1$  izanik. Demagun  $\delta$  erradioko  $p$ -ren ingurune batean  $g$ -ren hirugarren deribatua jarraitua dela, hots,  $\exists \delta > 0$ , zeinerako  $g \in C^3([p - \delta, p + \delta])$  den. Orduan,  $\exists \epsilon > 0$  zeinerako Steffensen-en metodoa konbergentea den  $\forall x_0 \in (p - \epsilon, p + \epsilon)$ . Gainera,  $|x_0^{(n)} - p| < \epsilon \forall n$  eta konbergentzia koadratikoa da.

## 3.6 Newton-Raphson Metodoa

Konbergentzia koadratikoko metodorik ezagunena da. Ondorioztatuko dugu puntu finkoko iteraziozko metodo koadratikotzat. Demagun  $f(x) = 0$  ekuazioaren soluzio bat  $p$  dela.  $g$  iteraziozko funtzio egoki bat aurkituko dugu. Horretarako,

$$g(x) = x + h(x)f(x)$$

idatziko dugu, eta orain  $h(x)$  hautatuko dugu halako moduan non  $g'(p) = 0$  den (apropoz, konbergentzia koadratikoa lortzeko). Horretarako, deriba dezagun:

$$g'(x) = 1 + h'(x)f(x) + h(x)f'(x)$$

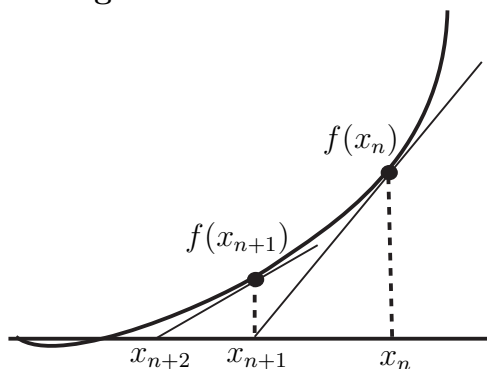
eta bereiziki,  $x = p$  hartuz,

$$g'(p) = 1 + h'(p) \overbrace{f(p)}^{=0} + h(p)f'(p) = 1 + h(p)f'(p) \implies h(p) = -\frac{1}{f'(p)}.$$

Azken hau lortzeko  $h(x) = -\frac{1}{f'(x)}$  hartuko dugu (nabaritu horretarako aukera bakarra ez dela). Ondorioz, Newton-Raphson-en formulara heldu gara:

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}$$

**Esanahi geometrikoa:**



- $(x_n, f(x_n))$ -tik igarotzen den zuzen ukitzalea:  $y - f(x_n) = f'(x_n) \cdot (x - x_n)$ .
- Zuzen ukitzalea eta  $y = 0$  ardatzarekiko ebakiduraren abszisa kalkulatuaz, Newton-en formula lortzen da.

Metodo honen makaltasun nabaria honako hau da:  $f'(x_n) = 0$  bada, ezin dugu zatitu!

**3.6 teorema** (N-R Konbergentzia lokaleko Teorema). Demagun  $f \in C^2([a, b])$  dela,  $\exists p \in [a, b]$ ,  $f(p) = 0$ ,  $f'(p) \neq 0$  izanik. Orduan,  $\exists \varepsilon > 0$  zeinerako Newton-Raphson-en metodoa konbergentea baita  $\forall x_0 \in (p - \varepsilon, p + \varepsilon)$ . Gainera,  $f \in C^3([a, b])$  bada, konbergentzia koadratikoa da.

*Frogapena:* Eginda dago,  $g'(p) = 0$  delako. ♣.

**3.7 teorema** (Newton Raphson metodoaren Konbergentzia globaleko Teorema). Demagun  $f \in C^2([a, b])$  dela, honako propietate hauek betetzen dituen:

1.  $f(a) \cdot f(b) < 0$ ;
2.  $f'(x) \neq 0 \quad \forall x \in [a, b]$ ;
3.  $f''(x) \neq 0 \quad \forall x \in [a, b]$ ;
4.  $\max \left\{ \left| \frac{f(a)}{f'(a)} \right|, \left| \frac{f(b)}{f'(b)} \right| \right\} \leq b - a$ .

Orduan,  $[a, b]$  tartean,  $\exists! p : f(p) = 0$  zeinerako Newton-Raphson-en metodoak  $p$ -rantz konbergitzen baitu  $\forall x_0 \in [a, b]$ . Gainera,  $f \in C^3([a, b])$  bada, konbergentzia koadratikoa da.

**3.4 adibidea** (Aplikazio bat). Bedi  $C > 0$ . Kalkulatuko dugu  $\sqrt[m]{C}$  Newton-Raphson metodoaren bariazio batez.

$$f(x) = x^m - C = 0$$

ekuazioa kontsideratuz, Newton-Raphson-en metodoari dagokion iteraziozko funtzioa:

$$\begin{aligned} g(x) &= x - \frac{f(x)}{f'(x)} = x - \frac{x^m - C}{mx^{m-1}} \\ &= \frac{mx^m - x^m + C}{mx^{m-1}} = \frac{(m-1)x^m + C}{mx^{m-1}} \\ &= \frac{m-1}{m}x + \frac{C}{mx^{m-1}} \implies \boxed{x_{n+1} = \frac{m-1}{m}x_n + \frac{C}{mx_n^{m-1}}} \end{aligned}$$

**Oharra:** Honela, bakarrik berretura bat (kasu honetan,  $m-1$  biderkaketa) kalkulatzen da, 2 berretura kalkulatu beharrean.

### 3.7 Ebakitzailleko metodoa

Newton-en metodoa “garestia” da (deribatuak kalkula eta ebaluatu behar dira). Makaltasun hori gainditzeko, formularen ordezkatuko dugu  $f'(x_n)$  (zuzen ukitzaillearen malda) zuzen ebakitzaille baten maldaz. Zehazkiago,

$$f'(x_n) = \lim_{x \rightarrow x_n} \frac{f(x_n) - f(x)}{x_n - x}$$

denez,  $|x_n - x_{n-1}|$  oso txikia bada, orduan

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

hurbilpena erabiliko dugu. Newton-Raphson-en formularen ordezkatuz,

$$\boxed{x_{n+1} = x_n - \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})} f(x_n)}$$

**Oharrak:**

- Teorikoki,  $x_{n+1} = \frac{x_{n-1}f(x_n) - x_n f(x_{n-1})}{f(x_n) - f(x_{n-1})}$  formula baliokidea da, baina ez dugu erabiliko desegonkorra delako:  $x_n$  eta  $x_{n-1}$   $p$ -ren hurbil badaude, orduan izendatzailea baita zenbakitzailea ere zeroren hurbil egongo dira, eta zenbakizko arazoak sor daitezke.
- Metodo hori ez da benetazko puntu finkoko iterazioa.
- Geometrikoki, baita analitikoki ere, *regula falsi* metodoaren oso antzekoa da, baina ebakitzaileko metodoan ez dira aztertzen zeinuak (horregatik sor daitezke arazoak,  $x_n$  eta  $x_{n+1}$  zeinu berekoak direnean).
- Metodoa aplikatzeko, pausu bakoitzean,  $f(x_n) \neq f(x_{n-1})$  nahitaezkoa da.
- Bi hasierako balioak dira beharrezkoak:  $x_0$  eta  $x_1$ .
- Metodoaren konbergentzia ez dago bermatuta orokorrean, baina honako teorema hau dugu:

**3.8 teorema.** Bedi  $f \in C^2([a, b])$  zeinerako  $\exists p \in [a, b] : f(p) = 0$  eta  $f'(p) \neq 0$  (hots, erro sinplea) baitira. Orduan,  $\exists \varepsilon > 0$  zeinerako  $\forall x_0, x_1 \in (p - \varepsilon, p + \varepsilon)$ , badaukagu

i)  $\{x_n\}_{n \geq 0}$  ebakitzaileko metodoaz sorturiko segida  $(p - \varepsilon, p + \varepsilon)$  tartean edukita dago.

ii)  $x_n \xrightarrow{k \rightarrow \infty} p$ .

Metodo horren konbergentziaren ordena urrezko zenbakia  $\Phi = \frac{1 + \sqrt{5}}{2} \simeq 1.61803 \dots$  da.

(ADIBIDEA)

## 3.8 Erro anitzkoitzak

**3.6 definizioa.** Esango dugu  $p \in \mathbb{R}$   $f$ -ren  $m$  ordenako erro anitzkoitza dela baldin eta

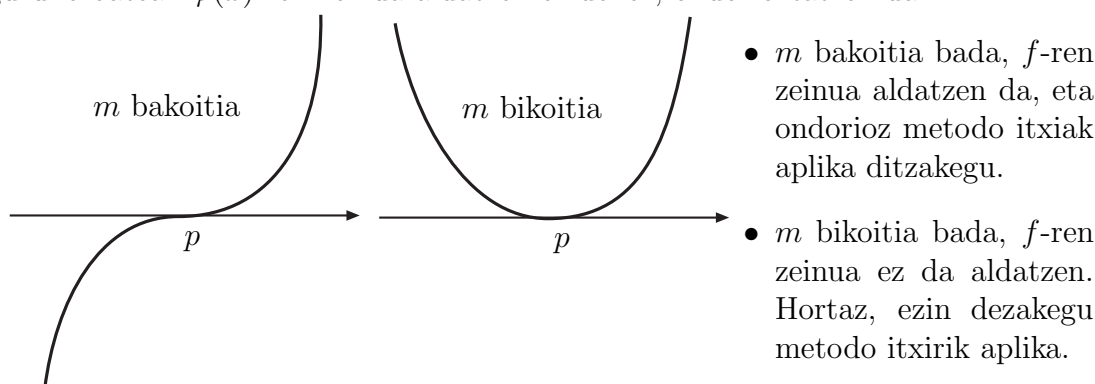
$$f(p) = f'(p) = f''(p) = \dots = f^{(m-1)}(p) = 0 \quad \text{eta} \quad f^{(m)}(p) \neq 0$$

bada. Esango dugu  $p$  erroaren anitzkoiztasuna  $m$  dela.

Bedi  $p$   $f$ -ren erro anitzkoitz bat,  $m$  ordenakoa. Taylor garapena erabiliz, badaukagu:

$$\begin{aligned} f(x) &= \sum_{i=m}^{\infty} \frac{f^{(i)}(p)}{i!} (x-p)^i \\ &= (x-p)^m \sum_{i=0}^{\infty} \frac{f^{(i+m)}(p)}{(i+m)!} (x-p)^i = (x-p)^m \varphi(x), \end{aligned}$$

non  $\varphi(p) \neq 0$  den.  $f(x) = (x-p)^m \varphi(x)$  adierazpena kontuan hartuz, eta  $p$ -ren ingurune batean  $\varphi(x)$ -ren zeinua aldatzen ez denez, ondorioztatzen da:



$p$  anitzkoitza bada,  $f'(p) = 0$  dugu (i.e. zuzen ukitsailea horizontala dugu!), eta orduan ezin dugu Newton-Raphson-en metodoa aplikatu konbergentzia koadratikoa lortu nahi badugu.  $p$ -ren hurbil,  $f'(x_n) \approx 0$  izango da, eta hortaz, zenbakizko arazoak sor daitezke. Aztertuko dugu N-R-en konbergentziaren ordena:

$$\begin{aligned} g(x) &= x - \frac{f(x)}{f'(x)} = x - \frac{(x-p)^m \varphi(x)}{m(x-p)^{m-1} \varphi(x) + (x-p)^m \varphi'(x)} \\ &= x - \frac{(x-p) \varphi(x)}{m \varphi(x) + (x-p) \varphi'(x)}, \end{aligned}$$

non  $\varphi(p) \neq 0$  dugun. Nabaritu azken izendatzailea  $\neq 0$  dela  $p$ -ren hurbil. Deribatuz,

$$\begin{aligned} g'(x) &= 1 - \frac{[\varphi(x) + (x-p)\varphi'(x)] \cdot [m\varphi(x) + (x-p)\varphi'(x)]}{[m\varphi(x) + (x-p)\varphi'(x)]^2} \\ &\quad + \frac{[m\varphi'(x) + \varphi'(x) + (x-p)\varphi''(x)] \cdot (x-p)\varphi(x)}{[m\varphi(x) + (x-p)\varphi'(x)]^2}. \end{aligned}$$

Orain,  $x = p$  hartuz,

$$g'(p) = 1 - \frac{m \cdot \varphi(p)^2}{m^2 \varphi(p)^2} = 1 - \frac{1}{m} \neq 0,$$

zeren eta  $m > 1$  den. Ondorioz, Newton-Raphson metodoa konbergentea izango da, baina  $m$  handia bada, konbergentzia motela izango da. Adibidez,  $m = 2$  bada, orduan  $g'(p) = 1/2$ , eta konbergituko du Bolzano metodoak bezain motel.

### Konbergentzia bizkortzeko era:

- $p$  erroaren anitzkoitzasuna  $m > 1$  bada, orduan  $g'(p) = 1 - 1/m < 1$  denez, erabil ditzakegu Aitken edo Steffensen metodoak (bigarrenaz, konbergentzia koadratikoa lortzen da).
- $p$  erroaren anitzkoitzasuna ezagutzen badugu (demagun  $m$  dela), orduan honako aldaketa hau egin dezakegu:

$$g(x) = x - m \frac{f(x)}{f'(x)} = \dots = x - m \frac{(x-p)\varphi(x)}{m\varphi(x) + (x-p)\varphi'(x)}.$$

Honela,  $g'(p) = 0$  (ariketa) eta konbergentzia koadratikoa lortuko genuke.

- $m$  ez badugu ezagutzen, kontsidera dezakegu:

$$\mu(x) = \frac{f(x)}{f'(x)} = \dots = \frac{(x-p)\varphi(x)}{m\varphi(x) + (x-p)\varphi'(x)}$$

funtzioa. Nabaritu  $\mu(p) = 0$ . Deribatuz,  $\mu'(p) = 1/m \neq 0$  lortzen da. Orduan,  $p$   $\mu$ -ren erro simple bat dugu, eta ondorioz, Newton-Raphson metodoa aplikatu diezaiokegu  $\mu$  funtzioari:

$$g(x) = x - \frac{\mu(x)}{\mu'(x)} = x - \frac{f(x)f'(x)}{(f'(x))^2 - f(x)f''(x)}.$$

### Oharrak:

- $\mu \in C^3([a, b]) \implies$  konbergentzia koadratikoa lortzen dugu.
- Metodo garestia da (3 ebaluaketa iterazio bakoitzean!)

(ADIBIDEA)





---

# 4. kapitulua

---

## Autobalio eta autobektoreen kalkulua

### 4.1 Sarrera

**4.1 definizioa.** Bedi  $\mathbb{K} = \mathbb{C}$  edo  $\mathbb{R}$ . Bitez  $\underline{x} \in \mathbb{K}^n$  bektore bat,  $A$  ( $n \times n$ ) dimentsioko matrize erreal bat eta  $\lambda \in \mathbb{K}$ ,

$$A \cdot \underline{x} = \lambda \cdot \underline{x}$$

egiaztatzen delarik. Orduan, esango dugu:

- $\underline{x}$  bektorea  $A$  matrizearen *autobektore* bat dela;
- $\lambda$  balioa  $A$  matrizearen *autobalio* bat dela,  $\underline{x}$  autobektoreari dagokiona.

$A$  matrizearen autobalio guztien (errealak baita konplexuak ere) multzoa  $\text{Spec}(A)$  notazioaz adieraziko dugu.

**4.1 teorema.**  $\lambda \in \mathbb{C}$   $A$ -ren autobalio bat da  $\iff p_A(\lambda) = 0$ , non  $p_A(t) = |A - tI_n|$  polinomio karakteristikoa baita.

**4.2 teorema.** Bedi  $A$  matrize simetriko bat. Orduan,

- $A$  matrize erreal bada  $\implies$  autobalio guztiak errealak dira;
- $\text{traza}(A) = \lambda_1 + \dots + \lambda_n$ ;
- $\det(A) = \lambda_1 \lambda_2 \dots \lambda_n$ .

**4.3 proposizioa.**  $A = (a_{ij})$  matrize triangeluar baten autobalioak  $a_{11}, a_{22}, \dots, a_{nn}$  dira.

*Frogapena:*  $|A - tI_n| = (a_{11} - t)(a_{22} - t) \dots (a_{nn} - t)$ . ♣

**Oharra:**  $\{\underline{v}_1, \dots, \underline{v}_n\}$   $\mathbb{R}^n$ -ren oinarri bat bada, orduan, sistema sortzaile denez,  $\underline{x} \in \mathbb{R}^n$  edozein bektore emanda,  $\exists \beta_1, \dots, \beta_n \in \mathbb{R}$  halako moduan non  $\underline{x} = \beta_1 \underline{v}_1 + \dots + \beta_n \underline{v}_n$  den.

## 4.2 Berreturako metodoa

Iteraziozko metodo bat da,  $A$ -ren autobektore bat kalkulatzeko, iteraziozko formula

$$\underline{x}^{(k+1)} = A \cdot \underline{x}^{(k)} \quad (4.1)$$

motakoa izanik.

### 4.2.1 Autobalio menperatzailearen kalkulua

Bedi  $A$  ( $n \times n$ ) dimentsioko matrize erreal bat beronen autobalioak  $\lambda_1, \lambda_2, \dots, \lambda_n$  direlarik,  $\underline{v}_1, \dots, \underline{v}_n$  autobektoreei dagozkienak, hurrenez hurren, i.e.

$$A \cdot \underline{v}_i = \lambda_i \underline{v}_i \quad \forall i = 1, \dots, n.$$

Suposa dezagun orain honako propietate hauek betetzen direla:

- i) Autobalio menperatzaile bat dago, hau da, orokortasuna galdu gabe,

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$$

betetzen dela. Kasu honetan, esango dugu  $\lambda_1$  autobalio menperatzailea dela.

- ii)  $\{\underline{v}_1, \dots, \underline{v}_n\}$  sistema linealki askea da (orduan,  $n$  luzerakoa izanik, oinarri bat da).

Orain (4.1) segidaren konbergentzia aztertuko dugu. Autobektoreen multzoa oinarri bat denez,  $\underline{x}^{(0)}$  edozein bektoretarako,  $\exists \alpha_1, \dots, \alpha_n \in \mathbb{R}$  non

$$\underline{x}^{(0)} = \sum_{i=1}^n \alpha_i \underline{v}_i$$

den. Ondorioz, linealtasuna erabiliz,

$$A^k \underline{x}^{(0)} = A^k \left( \sum_{i=1}^n \alpha_i \underline{v}_i \right) = \sum_{i=1}^n \alpha_i A^k \cdot \underline{v}_i = \sum_{i=1}^n \alpha_i \lambda_i^k \underline{v}_i, \quad (4.2)$$

non

$$A^k \underline{v}_i = A^{k-1} \cdot A \underline{v}_i = A^{k-1} \lambda_i \underline{v}_i = \lambda_i A^{k-1} \underline{v}_i = \dots = \lambda_i^k \underline{v}_i$$

erabili dugun (beraz,  $\text{Spec}(A^k) = \{\lambda_1^k, \dots, \lambda_n^k\}$  dela frogatu dugu). Orain, (4.2) formula autobalio menperatzaileaz  $k$  aldiz biderkatu eta zatituz:

$$A^k \underline{x}^{(0)} = \lambda_1^k \sum_{i=1}^n \alpha_i \left( \frac{\lambda_i}{\lambda_1} \right)^k \underline{v}_i = \lambda_1^k \left[ \alpha_1 \underline{v}_1 + \sum_{i=2}^n \alpha_i \overbrace{\left( \frac{\lambda_i}{\lambda_1} \right)^k}^{|\cdot| < 1} \underline{v}_i \right].$$

Ondorioz,

$$\lim_{k \rightarrow \infty} (A^k \underline{x}^{(0)}) = \lim_{k \rightarrow \infty} \lambda_1^k \alpha_1 \underline{v}_1.$$

Hortaz,  $\alpha_1 \neq 0$  bada, azter dezakegu azken limite honen konbergentzia  $\lambda_1$ -en arabera:  $|\lambda_1| < 1$  bada,  $\implies$  konbergentea da zerorantz, eta  $|\lambda_1| \geq 1$  bada,  $\implies$  ez du konbergitzen (edo auskalo).

Orain, osagai osagai zatituz, (zehatzago,  $\underline{x}^{(k+1)}$  eta  $\underline{x}^{(k)}$  (4.1) segidaren bektoreen  $i$ -garren osagaiak zatituz)

$$\frac{(\underline{x}^{(k+1)})_i}{(\underline{x}^{(k)})_i} = \lambda_1 \frac{\left[ \alpha_1 \underline{v}_1 + \sum_{j=2}^n \alpha_j \left( \frac{\lambda_j}{\lambda_1} \right)^{k+1} \underline{v}_j \right]_i}{\left[ \alpha_1 \underline{v}_1 + \sum_{j=2}^n \alpha_j \left( \frac{\lambda_j}{\lambda_1} \right)^k \underline{v}_j \right]_i}$$

Hortaz,  $k \rightarrow \infty$  limitea hartuz, ondorioztatzen da

$$\lim_{k \rightarrow \infty} \frac{(A^{k+1} \underline{x}^{(0)})_i}{(A^k \underline{x}^{(0)})_i} = \lambda_1, \quad \alpha_1 \neq 0 \text{ bada.} \quad (4.3)$$

(4.1) segida dibergentea bada, bi osagaien zatiduran overflow gerta daiteke. Horregatik, (4.1) segida aldatuko dugu, pausu bakoitzean lortzen den bektorea “normalizatuz”.

**ALGORITMOA:**

$\underline{x}^{(0)} \in \mathbb{R}$ ;  $\|\underline{x}^{(0)}\|_\infty = 1$  emanda,  $k = 0, 1, 2, \dots$  bada,

1.  $\underline{y}^{(k+1)} = A \cdot \underline{x}^{(k)}$  kalkulatu;
2.  $\mu^{(k+1)} = y_p^{(k+1)}$ , non  $\|\underline{y}^{(k+1)}\|_\infty = |y_p^{(k+1)}| = \max_{i=1, \dots, n} |y_i^{(k+1)}|$  den ;
3. “Normalizatu”  $\underline{y}^{(k+1)}$  honela:  $\underline{x}^{(k+1)} = \frac{\underline{y}^{(k+1)}}{\mu^{(k+1)}}$ .
4. Errepikatu 1. – 3. pausuak behin eta berriro.

Honela, ikus daiteke  $\{\underline{x}^{(k)}\} \xrightarrow{k \rightarrow \infty} \underline{v}_1$  eta  $\{\mu^{(k)}\} \xrightarrow{k \rightarrow \infty} \lambda_1$  lortzen dela. Beraz,  $k$  haundia denean, erraz ikus daiteke (ariketa) konbergitzekotan, zenbait  $i$ -tarako,

$$\mu^{(k+1)} = \frac{(A^{k+1} \underline{x}^{(0)})_i}{(A^k \underline{x}^{(0)})_i}$$

egiaztatzen dela. Ondorioz, (4.3) erabiliz,  $\mu^{(k)} \xrightarrow{k \rightarrow \infty} \lambda_1$  lortzen da.

#### Oharrak:

- **Gelditze-irizpidea(k)** Zehaztasuna azter daiteke autobektorean edo autobalioan. Autobektorean, adibidez,  $\|\underline{x}^{(k+1)} - \underline{x}^{(k)}\|_\infty \leq \varepsilon$  denean, algoritmoaren irteera izango da  $\underline{x}^{(k+1)} \approx \underline{v}_1$  eta  $\mu^{(k+1)} \approx \lambda_1$ . Irizpidea izan daiteke  $|\mu^{(k+1)} - \mu^{(k)}| \leq \varepsilon$  izatea, baita ere errore erlatiboan.
- Eragozpenak: metodoak funtzionatzeko beharrezkoa da  $\alpha_1 \neq 0$  izatea, baita  $\lambda_1$  autobalio erreal eta simple izatea, ere.
- Eragozpenik ez badaude, konbergentzia bermatuta dago, eta lineala da  $\implies$  bizkor dezakegu konbergentzia Aitken  $\Delta^2$  metodoaz, autobalioan.

(ADIBIDEA)

## 4.2.2 Modulo txikieneko autobalioaren kalkulua

### Berretura alderantzizkoko metodoa

Demagun  $A$ -ren autobalioak

$$\lambda_1, \lambda_2, \dots, \lambda_n$$

direla, non:

$$|\lambda_n| < |\lambda_{n-1}| \leq \dots \leq |\lambda_1|$$

den.  $A\tilde{v}_i = \lambda_i\tilde{v}_i \implies (1/\lambda_i)\tilde{v}_i = A^{-1}\tilde{v}_i$ enez,  $A^{-1}$  matrizearen autobalioak

$$\frac{1}{\lambda_n}, \frac{1}{\lambda_{n-1}}, \dots, \frac{1}{\lambda_1}$$

ditugu,

$$\left| \frac{1}{\lambda_n} \right| > \left| \frac{1}{\lambda_{n-1}} \right| \geq \dots \geq \left| \frac{1}{\lambda_1} \right|$$

egiaztatzen delarik. Ondorioz,  $\frac{1}{\lambda_n}$  autobalioa  $A^{-1}$  matrizearen autobalio menperatzaileaenez, berreturako metodoa  $A^{-1}$  matrizearekin erabiltzen badugu, irteera  $1/\lambda_n$ -ren hurbilpen on bat izango litzateke, eta horren alderantzizkoa  $\lambda_n$ -ren hurbilpen on bat izango da. Horretarako, iterazio bakoitzean,

$$\tilde{y}^{(k+1)} = A^{-1} \cdot \tilde{x}^{(k)}$$

kalkulatu beharrean,

$$A \cdot \tilde{y}^{(k+1)} = \tilde{x}^{(k)}$$

sistema ebatziko dugu,  $A^{-1}$  matrizearen kalkulua saihesteko. Beraz,  $A$  matrizeaz definituriko sistema asko ebatzi behar dugunez, ezabapen trinkoaren metodoa erabili dugu, hasieran  $A = L \cdot U$  faktorizazioa lortuz (Doolittle edo Crout metodoaz). Nabaritu  $\tilde{v}_n$  bektorea  $A^{-1}$ -en  $1/\lambda_n$  autobalioari dagokion autobektorea dela (metodoaz

kalkulaturikoa).

[ADIBIDEA]

### 4.2.3 Berretura alderantzizkoko metodo lerratua

Berretura alderantzizkoko metodoa oso egokia da edozein autobalioaren hurbilpen fidagarri bat ezagutzen badugu. Demagun  $\mu \approx \lambda_i$  hurbilpen fidagarri bat ezagutzen dugula. Orduan,  $A - \mu I$  matrizea aintzakotzat hartuz, erraz ikus daiteke

$$\text{Spec}(A - \mu I) = \{(\lambda_1 - \mu), (\lambda_2 - \mu), \dots, (\lambda_n - \mu)\}$$

dela (beraz,  $\tilde{v}_1, \dots, \tilde{v}_n$  dira, hurrenez hurren, dagozkien autobektoreak). Orain,

berehalakoa da  $A - \mu I$  matrizearen modulu txikieneko autobalioa  $\lambda_i - \mu$  dela,  $\mu \approx \lambda_i$  izateagatik. Hortaz, berretura alderantzizkoko metodoa aplikatzen badiogu  $A - \mu I$  matrizeari,  $\lambda_i - \mu$  balioaren hurbilpena lortuko genuke azken irteeratzat. Lorturiko emaitzari  $\mu$  gehituz,  $\lambda_i$  kalkulatu nahi genuen balioa lortzen da.

[ADIBIDEA]



---

# 5. kapitulua

---

## Interpolazioa

### 5.1 Sarrera

IDEIA:  $f: \mathbb{R} \rightarrow \mathbb{R}$  funtzio bat ordezkatu nahi dugu  $\bar{f}$  beste funtzio batez, non

- $\bar{f}$  “erabilgarriagoa” den (beraz,  $\bar{f}$  polinomio bat izango da);
- $\bar{f}$  funtzioa  $f$ -rekin bat dator puntu jakin batzuetan.

### 5.2 Polinomio interpolatzailea. Lagrange metodoa

ARAZO OROKORRA:  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$  puntuak emanda, aurkitu  $n$  mailako  $P(x)$  polinomio bat,  $P(x_0) = y_0, \dots, P(x_n) = y_n$  egiaztatzen duena.

PAUSUZ PAUSU:

$n = 1$  Kontsideratu  $(x_0, y_0)$  eta  $(x_1, y_1)$  datuak. Puntu haietatik igarotzen den zuzena (hots, 1-mailako polinomioa) honako hau da:

$$P(x) = \frac{x - x_1}{x_0 - x_1} \cdot y_0 + \frac{x - x_0}{x_1 - x_0} \cdot y_1$$

Beraz,

$$P(x_0) = 1 \cdot y_0 + 0 \cdot y_1 = y_0; \quad P(x_1) = 0 \cdot y_0 + 1 \cdot y_1 = y_1.$$

$n = 2$  Kontsideratu, orain, hiru puntu:  $(x_0, y_0), (x_1, y_1)$  eta  $(x_2, y_2)$ . Puntu haietatik igarotzen den parabola (hots, 2-mailako polinomioa) honako hau da:

$$P(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} \cdot y_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} \cdot y_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \cdot y_2$$

Beraz,

$$P(x_0) = 1 \cdot y_0 + 0 \cdot y_1 + 0 \cdot y_2; \quad P(x_1) = 0 \cdot y_0 + 1 \cdot y_1 + 0 \cdot y_2; \quad P(x_2) = 0 \cdot y_0 + 0 \cdot y_1 + 1 \cdot y_2.$$

Orokorrean, honako teorema hau daukagu:

**5.1 teorema.** Bitez  $n \in \mathbb{N}$  eta  $x_0, x_1, \dots, x_n$  abszisa desberdinak. Orduan,  $y_0, \dots, y_n$  balioak emanda, existitzen da  $P$  polinomio bakar bat  $P(x_0) = y_0, \dots, P(x_n) = y_n$  eta  $\text{maila}(P) \leq n$  egiaztatzen dituen.

*Frogapena:*

Existentzia: Lagrange funtzio oinarrikoak eraikiko ditugu:

$$L_j(x) = \prod_{i=0, i \neq j}^n \frac{x - x_i}{x_j - x_i} \implies L_j(x_i) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases}$$

Hemendik abiatuz, erraz lortzen da nahi dugun polinomioa

$$P(x) = y_0 \cdot L_0(x) + y_1 \cdot L_1(x) + \dots + y_n \cdot L_n(x).$$

Bakartasuna: Suposatu  $P_1(x)$  eta  $P_2(x)$  teoremaren baldintzak betetzen dituzten bi polinomio direla (i.e.,  $P_1(x_i) = P_2(x_i) = y_i \quad \forall i = 0, \dots, n$ , eta  $\text{maila}(P_1), \text{maila}(P_2) \leq n$ ). Orduan,  $P(x) = P_1(x) - P_2(x)$  polinomioak honako propietate hauek betetzen ditu:

i)  $P(x_i) = 0 \quad \forall i = 0, \dots, n$  ( $\implies n + 1$  erro desberdin ditu  $P$ -k);

ii)  $\text{maila}(P) \leq n$ .

Ondorioz,  $P(x) \equiv 0 \implies P_1(x) = P_2(x) \quad \forall x \in \mathbb{R}$ . ♣.

**Oharrak:**

- Esango dugu  $P$   $f$ -ren interpolazio-polinomioa dela,  $x_0, \dots, x_n$  nodoekikoa.
- Nodo bat aldatzen bada, edo gehitzen bada, orduan  $L_j$  guztiak aldatzen dira osoz.
- Interpolazio-polinomioa kalkulatzeko arazoa txarto baldintzatua dago.

**5.2 teorema.** Bedi  $f \in C^{n+1}([a, b])$ , eta bedi  $P(x)$   $f$ -ren interpolazio-polinomioa  $x_0, \dots, x_n \in [a, b]$  nodoetarako. Orduan,  $\forall x \in [a, b]$  existitzen da  $\xi \in (a, b)$  zeinerako

$$e(x) = f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi(x)$$

baita, non  $\Pi(x) = (x - x_0)(x - x_1) \dots (x - x_n)$  den.



**Oharrak:**

- $\xi = \xi_x$ ;
- $f^{(n+1)}(x)$  bornatuta dago  $[a, b]$  tartean jarraitua delako. Beraz,  $M_{n+1} = \max_{\xi \in [a, b]} \{|f^{(n+1)}(x)|\}$  hartuz,

$$|e(x)| \leq \frac{M_{n+1}}{(n+1)!} |\Pi(x)| \leq \frac{M_{n+1}}{(n+1)!} (b-a)^{n+1}$$

dugu. Ondorioz, bornaketa orokor bat existitzen bada, hau da, existitzen bada  $A \in \mathbb{R}$  zeinerako  $M_n \leq A \quad \forall n \in \mathbb{N}$  betetzen baita, orduan

$$P_n(x) \xrightarrow{k \rightarrow \infty} f(x) \quad \forall x \in (a, b).$$

(ADIBIDEA)

### 5.3 Newtonen diferentzia zatituen metodoa

IDEIA: Lagrange metodoaz, nodo berri bakoitza gehitzen denean, kalkulu guztiak egin behar dira berriro. Newton metodoaz, makaltasun hori saihestuko dugu.

Demagun  $q_k(x)$   $f$ -ren interpolazio-polinomioa dela  $x_0, \dots, x_k$  nodoetarako (adibidez,  $q_0(x) = f(x_0)$ ). Orduan,

$$q_k(x) = q_{k-1}(x) + h_k(x)$$

dugu, non maila( $h_k$ )  $\leq k$  den. Orduan,

$$h_k(x_i) = q_k(x_i) - q_{k-1}(x_i) = 0 \quad \forall i = 0, 1, \dots, k-1.$$

eta hortaz,

$$h_k(x_i) = a_k(x - x_0)(x - x_1) \dots (x - x_{k-1}). \quad (5.1)$$

Orain, aurkituko dugu  $a_k \in \mathbb{R}$  koefizientea. Horretarako, (5.1) ekuazioan bakanduz eta  $x = x_k$  hartuz,

$$a_k = \frac{q_k(x_k) - q_{k-1}(x_k)}{(x_k - x_0) \dots (x_k - x_{k-1})} = \frac{f(x_k) - q_{k-1}(x_k)}{(x_k - x_0) \dots (x_k - x_{k-1})} \quad (5.2)$$

lortzen da.  $k = 0$ -tik hasiz,

$$q_0(x) = f(x_0)$$

$$q_1(x) = f(x_0) + a_1(x - x_0)$$

$$q_2(x) = f(x_0) + a_1(x - x_0) + a_2(x - x_0)(x - x_1)$$

⋮

$$P(x) = q_n(x) = f(x_0) + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0) \dots (x - x_{n-1}).$$

Nabaritu metodo honen bidez, egindako kalkulu guztiak aprobeztatzen direla nodo berri bat gehitzen den bakoitzean!

**5.1 definizioa.**  $a_k$  (5.2) ekuazioak definituriko koefizienteari  $k$  ordenako diferentzia zatitua deituko diogu, eta

$$a_k = f[x_0, x_1, \dots, x_k]$$

notazioaz adieraziko dugu.

### 5.3.1 Diferentzia zatituen propietateak

1.  $a_k$  diferentzia zatitua  $q_k(x)$  polinomioan  $x^k$  berreturarekin doan koefizientea da.
2. Nodoen ordenak ez dauka eraginik  $a_k$  balioan. Zehazki,  $\sigma \{0, 1, \dots, k\}$  multzoaren permutazio bat bada, orduan

$$f[x_0, \dots, x_k] = f[x_{\sigma(0)}, \dots, x_{\sigma(k)}].$$

3.

$$f[x_0, x_1, \dots, x_k] = \frac{f[x_1, x_2, \dots, x_k] - f[x_0, x_1, \dots, x_{k-1}]}{x_k - x_0} \quad \forall k \geq 1$$

### 5.3.2 Diferentzia zatituak kalkulatzeko algoritmoa

(TAULA: ORRIA)

$$\begin{aligned} P(x) = & f[x_0] + f[x_0, x_1](x - x_0) \\ & + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ & + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \\ & + \dots \\ & + f[x_0, x_1, \dots, x_n](x - x_0) \dots \dots \dots (x - x_{n-1}) \end{aligned}$$

polinomioa balioztatzeko era optimoa Hörner teknikaren honako bariazio hau da:

$$P(x) = [\dots [[a_n(x - x_{n-1}) + a_{n-1}](x - x_{n-2}) + a_{n-2}] \dots + a_1](x - x_0) + a_0$$

ALGORITMOA:

1.  $b_n = a_n$ ;
2.  $b_i = b_{i+1}(x - x_i) + a_i \quad i = n - 1, n - 2, \dots, 0$
3.  $b_0 = P(x)$  (irteera).

### 5.3.3 Memorian biltegitraketa

Hasierako datuak (hori da,  $(x_0, f(x_0)), \dots, (x_n, f(x_n))$ ) bikoteak) biltegitratzeko,  $n+1$  luzerako bi bektore behar izaten dira:

$$\begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \quad \text{eta} \quad \begin{pmatrix} f(x_0) \\ f(x_1) \\ \vdots \\ f(x_n) \end{pmatrix}$$

Ikusiko dugunez, bi bektore horiekin nahikoa dugu polinomioaren koefizienteak kalkulatzeko. Nodoen bektorea ez da aldatuko (diferentzia bakoitza kalkulatzeko beharrezkoa izango baita), eta bestea aldatuko da behetik gora:  $f[x_n]$  eta  $f[x_{n-1}]$  erabiliz kalkulatu da  $f[x_{n-1}, x_n]$  diferentzia. Bestalde,  $f[x_n]$  gehiago erabiltzen ez denez, gainean idatziko dugu kalkulaturiko balio berria:

$$\begin{pmatrix} f[x_0] \\ f[x_1] \\ \vdots \\ f[x_{n-1}] \\ f[x_n] \end{pmatrix} \quad \longrightarrow \quad \begin{pmatrix} f[x_0] \\ f[x_1] \\ \vdots \\ f[x_{n-1}] \\ f[x_{n-1}, x_n] \end{pmatrix}$$

Orain, gauza bera egingo dugu hurrengo bi osagaiekin, eta hori errepikatuko dugu bigarren ordenako diferentzia guztiak kalkulatu arte, gehiago erabiltzen ez ditugun gaien gainean berriak idatziz:

$$\longrightarrow \begin{pmatrix} f[x_0] \\ f[x_1] \\ \vdots \\ f[x_{n-3}] \\ f[x_{n-2}, x_{n-1}] \\ f[x_{n-1}, x_n] \end{pmatrix} \quad \longrightarrow \dots \longrightarrow \begin{pmatrix} f[x_0] \\ f[x_0, x_1] \\ f[x_1, x_2] \\ \vdots \\ f[x_{n-2}, x_{n-1}] \\ f[x_{n-1}, x_n] \end{pmatrix}$$

eta era berean jarraituko dugu behetik gora, ondorengo ordenako diferentziak kalkulatu eta biltegitratuz, ordena guztiekin, amaitu arte

$$\longrightarrow \dots \longrightarrow \begin{pmatrix} f[x_0] \\ f[x_0, x_1] \\ f[x_0, x_1, x_2] \\ \vdots \\ f[x_{n-2}, x_{n-1}, x_n] \end{pmatrix} \quad \longrightarrow \dots \longrightarrow \begin{pmatrix} f[x_0] \\ f[x_0, x_1] \\ f[x_0, x_1, x_2] \\ \vdots \\ f[x_0, x_1, \dots, x_n] \end{pmatrix}$$

Azkenean, bigarren bektorean  $P(x)$  polinomioaren koefizienteak izango ditugu.

### 5.3.4 Motze-errorearen azterketa

Kontsideratu  $(x_0, f(x_0), \dots, (x_n, f(x_n)))$  nodo desberdinak, dagozkien balioekin batera, eta kontsideratu  $P_n$  haietatik igarotzen den interpolazio-polinomioa ( $f$ -rekin bat datorrena adierazitako nodoetan). Egindako errorea azpiko hau da (gogoratu 5.2 teorema):

$$R_n(x) = e(x) = f(x) - P_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \Pi(x),$$

non  $\xi \in [x_0, x_n]$ .

**5.3 teorema.**  $x_0, \dots, x_n$  nodoak emanda, eta  $P_n$  polinomioa kontsideratuz, badaukagu

$$R_n(x) = f(x) - P_n(x) = f[x_0, \dots, x_n, x] \Pi(x) \quad \forall x \neq x_0, \dots, x_n.$$

*Frogapena:* Kontsideratu  $q(t)$  interpolazio-polinomioa  $\{x_0, \dots, x_n, x\}$  nodoetarako (kontsideratzen ari gara  $x$  nodotzat). Orduan,

$$q(t) = P_n(t) + f[x_0, \dots, x_n, x] \overbrace{(t-x_0) \dots (t-x_n)}^{\Pi(t)}$$

Ondorioz,  $t = x$  hartuz, badaukagu

$$f(x) = q(x) = P_n(x) + f[x_0, \dots, x_n, x] \Pi(x),$$

$x$  nodo bat delako. Amaitzeko,

$$R_n(x) = f(x) - P_n(x) = f[x_0, \dots, x_n, x] \Pi(x)$$



Orain, 5.2 eta 5.3.4 teoremak erabiliz, ondoriozta dezakegu diferentzia zatituen honako propietate hau:

**5.4 teorema.** Bedi  $f \in C^n([a, b])$ , eta bitez  $x_0, x_1, \dots, x_n \in [a, b]$  nodo desberdinak. Orduan,

$$\exists \xi \in (a, b) : \frac{f^{(n)}(\xi)}{n!} = f[x_0, x_1, \dots, x_n]$$

**Oharrak praktikarako:**

- Teorema 5.2 erabili ohi da errorea bornatzeko  $f^{(n+1)}$ -en bornaketa ezagutzen den kasuan.
- Teorema 5.3 erabiltzen da errorea estimatzeko,  $(x_{n+1}, f(x_{n+1}))$  beste nodo bat ezagutzen dugunean,  $f[x_0, \dots, x_n, x] \approx f[x_0, \dots, x_n, x_{n+1}]$  estimazioa hartuz.

(ADIBIDEA)

**5.1 adibidea.**  $f$  funtzioa  $n$  mailako polinomioa den kasuan, badaukagu  $f(x) = P_n(x) \quad \forall x \in \mathbb{R}$ , eta ondorioz,  $0 = R_n(x) = R_{n+1}(x) = R_{n+2}(x) = \dots$ . Egoera hori nabaritzen da taulan:  $f[x_0, \dots, x_n] = f^{(n)}(\xi_x)/n!$ , baina  $f^{(n)} \equiv \text{konstante}$  da  $\implies f[x_0, \dots, x_{n+1}] = 0$ .

$i$	$x_i$	$f[x_i]$	$f[x_i, x_j]$	$f[x_i, x_j, x_k]$	$f[x_i, \dots]$	$f[x_i, \dots]$	$f[x_i, \dots]$
0	0	-5					
1	1	1	6	2			
2	3	25	12	6	1	0	
3	4	55	30	11	1	0	0
4	6	181	63	15	1		
5	7	289	108				

Ikusten denez,  $f$  funtzioa oso ondo hurbil daiteke 3.mailako polinomio batez. Nabaritu azken honek ez du esan nahi beraz  $f$  polinomio bat dela.

## 5.4 Nodo distantzikideak: Diferentzia finituak

Demagun  $x_0, \dots, x_n$  nodoak distantzikideak direla, hau da, existitzen dela  $h > 0$  halako moduan non

$$x_1 - x_0 = x_2 - x_1 = \dots = x_n - x_{n-1} = h$$

den. Beste erara esanda,

$$x_j = x_0 + j \cdot h \quad \forall j = 0, \dots, n.$$

Kasu honetan, diferentzia zatituen metodoen kalkuluak asko sinplifikatzen dira.

**5.2 definizioa. Aurrerakako diferentzia finituak** ( $h$  finkatuta dago)

$$\begin{aligned} \Delta^0 f(x) &= f(x) \\ \Delta f(x) &= f(x+h) - f(x) \\ \Delta^2 f(x) &= \Delta(\Delta f(x)) &= \Delta f(x+h) - \Delta f(x) \\ \Delta^3 f(x) &= \Delta(\Delta^2 f(x)) &= \Delta^2 f(x+h) - \Delta^2 f(x) \\ &\vdots \\ \Delta^n f(x) &= \Delta(\Delta^{n-1} f(x)) &= \Delta^{n-1} f(x+h) - \Delta^{n-1} f(x) \end{aligned}$$

*Diferentzia finituen taula* askoz errazago eraikitzen da (zatitu gabe kalkulatzeko dira), azpiko adibide honetan ikus dezakegunez:

$i$	$x_i$	$f(x_i)$	$\Delta(f)$	$\Delta^2(f)$	$\Delta^3(f)$	$\Delta^4(f)$
0	0	-5				
			6			
1	1	1		2		
			8		6	
2	2	9		8		0
			16		6	
3	3	25		14		
			30			
4	4	55				

Polinomio interpolatzailea eraikitzeko, honako formula hauek hartuko ditugu aintzakotzat:

$$\begin{aligned}
 f[x_0, x_1] &= \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{\Delta f(x_0)}{h} \\
 f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\Delta f(x_0 + h)/h - \Delta f(x_0)/h}{2h} = \frac{\Delta^2 f(x_0)}{2h^2} \\
 &\vdots
 \end{aligned}$$

Orokorrean, lortzen da honako formula hau:

$$\boxed{f[x_0, x_1, \dots, x_n] = \frac{\Delta^n f(x_0)}{n!h^n}} \quad (5.3)$$

Azken formula hori erabiliko dugu aurreko ataleko Newton polinomio interpolatzailearen formulak diferentzia finituen hizkuntzara "itzultzeko".  $P_n(x)$  polinomioa idazteko,  $x = x_0 + \alpha \cdot h$  aldagai aldaketa egingo dugu, non  $\alpha$  aldagai berria den. Alde batetik, badaukagu

$$x - x_i = x_0 + \alpha \cdot h - (x_0 + i \cdot h) = h \cdot (\alpha - i)$$

eta, hortaz,

$$(x - x_0) \dots (x - x_k) = h^{k+1} \cdot \alpha \cdot (\alpha - 1) \dots (\alpha - k).$$

Goiko formula hauek erabiliz, ondorioztatzen da zuzenean

$$\begin{aligned}
 P_n(x_0 + \alpha \cdot h) &= f(x_0) + \alpha \Delta f(x_0) \\
 &+ \alpha(\alpha - 1) \frac{\Delta^2 f(x_0)}{2!} \\
 &+ \alpha(\alpha - 1)(\alpha - 2) \frac{\Delta^3 f(x_0)}{3!} \\
 &\dots \\
 &+ \alpha(\alpha - 1) \dots (\alpha - n + 1) \frac{\Delta^n f(x_0)}{n!}
 \end{aligned}$$

(5.3) formula erabiliz, erraz ondorioztatzen dira egindako errorearen adierazpen zehatza eta hurbildua:

$$\begin{aligned}
 R_n(x) &= h^{n+1} \alpha(\alpha - 1) \dots (\alpha - n) \frac{f^{(n+1)}(\xi_x)}{(n+1)!}, \quad \xi_x \in (x_0, x_n) \\
 R_n(x) &\approx \alpha(\alpha - 1) \dots (\alpha - n) \frac{\Delta^{n+1} f(x_0)}{(n+1)!}
 \end{aligned}$$

Nabaritu  $f$ -ren beste nodo bateko balioa ezagutu behar dugula azken estimazioa lortzeko.

[ADIBIDEA]

### 5.4.1 Errorearen zabaltzea

- Diferentzia finituen tauletan errorea era triangeluarrean zabaltzen da.

(ADIBIDEA)

- Normalean, diferentzia finituen tauletan,  $f$  “ona” bada (diferentziagarria, erregular...), diferentziak beherakorrak dira (ordenarekiko), eta berdinak izateko joera dute, polinomio batean gertatzen den bezala.
- Hots, zeinua alternatiboki aldatzen bada zutabeetan, edo diferentziaren modua gorakorra bada, orduan esan dezakegu errore bat gertatu dela (borobiltze-errorea, edo beste motakoa). Zutabe horiek “susmagarriak” izango dira, eta horregatik ez ditugu erabiliko interpolatzeko.

**Adibidea:**

*Non dago  
errorea?*

$x_i$	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$	$\Delta^4 f(x_i)$	$\Delta^5 f(x_i)$
1	1					
2	4	3				
3	9	5	2	0		
4	16	7	2	0	0	-1
5	25	9	2	-1	-1	5
6	35	10	1	3	4	-10
7	49	14	4	-3	-6	10
8	64	15	1	1	4	-5
9	81	17	2	0	-1	
10	100	19	2			



## 5.5 Hermite polinomioa

Interpolazioaren jatorriko problemaren bariazio bat kontsideratuko dugu. Bitez  $x_0, \dots, x_n \in [a, b]$  nodo desberdinak, eta  $f \in C^1([a, b])$ . Orduan, aurkitu nahi dugu  $P(x)$  polinomio bat, azpiko baldintza hauek betetzen dituen:

- $\text{maila}(P) \leq 2n + 1$ ;
- $P(x_i) = f(x_i), \quad \forall i = 0, \dots, n$ ;
- $P'(x_i) = f'(x_i), \quad \forall i = 0, \dots, n$ .

Existitzekotan,  $P(x)$  polinomioari *Hermite polinomio (oskulatzailea)* deituko diogu.

**5.5 teorema.** Bitez  $x_0, \dots, x_n \in [a, b]$  nodo desberdinak, eta  $f \in C^1([a, b])$ . Orduan,  $f$  funtziorako, Hermite polinomioa existitzen da eta bakarra da.

**5.6 teorema.** Bedi  $f \in C^{2n+2}([a, b])$  eta bedi  $P(x)$  bere Hermite polinomioa  $x_0, \dots, x_n \in [a, b]$  nodo desberdinetarako. Orduan,  $\forall x \in [a, b]$ , badaukagu

$$\exists \xi_x \in (a, b) : \quad e(x) = f(x) - P(x) = \frac{f^{2n+2}(\xi_x)}{(2n+2)!} \Pi^2(x),$$

non  $\Pi^2(x) = (\Pi(x))^2 = (x - x_0)^2(x - x_1)^2 \dots (x - x_n)^2$  den.

### Oharrak:

- Interpolazio arruntean bezala,  $f$ -ren deribatu guztiek borne komun badute, orduan, erroreak zerorantz jotzen du  $n \rightarrow \infty$  doanean.
- Errorearen adierazpenari begira, Hermite polinomioa interpreta dezakegu polinomio interpolatzaile arrunt bat bezala nodoak errepikatzen direnean.

Hermite polinomioa eraikitzeke, Newton diferentzia zatituen teknikaren ondorengo bariazioa erabiliko dugu, azken oharra kontuan hartuz: Kontsideratuko ditugu ondorengo nodoak:

$$x_0, x_0, x_1, x_1, x_2, x_2, \dots, x_n, x_n$$

interpolaziorako datuak ondorengoak izanik:

$$f(x_0), f'(x_0), f(x_1), f'(x_1), \dots, f(x_n), f'(x_n).$$

Orain, nodoak berrizendatuko ditugu ondorengo erara:

$$z_0 = x_0, z_1 = x_0, z_2 = x_1, z_3 = x_1, \dots, z_{2n} = x_n, z_{2n+1} = x_n.$$

Hermite polinomioaren adierazpena honako hau da:

$$P(x) = f[z_0] + \sum_{i=1}^{2n+1} f[z_0, z_1, \dots, z_i](x - z_0) \dots (x - z_{i-1}).$$

Horretarako, diferentzia zatituen taula eraiki eta Newton-en formula aplikatu behar dugu zuzenean. Diferentzia zatituak kalkulatzeko sortuko den arazo bakarra lehen ordenako diferentzietan dago:

- $f[z_{2i-1}, z_{2i}]$  motako diferentziak problemarik gabe kalkulatu dira:

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

- Bestalde,  $f[z_{2i}, z_{2i+1}]$  motako diferentziak zuzenean kalkulatu nahi badugu:

$$f[z_{2i}, z_{2i+1}] = \frac{f(z_{2i+1}) - f(z_{2i})}{z_{2i+1} - z_{2i}} = \frac{f(x_i) - f(x_i)}{x_i - x_i} = \frac{0}{0}$$

Zailtasun hori saihesteko, kontuan hartuko dugu 5.3 teorema:

$$\exists \xi \in [x_0, x_i] : f[x_0, \dots, x_i] = \frac{f^{(i)}(\xi)}{i!} \quad (5.4)$$

Nabaritu nodoak ordenatuta daudela suposatu dugula. (5.4) erabiliz, badaukagu

$$f[z_{2i}, z_{2i+1}] = f[x_i, x_i] = \frac{f'(x_i)}{1!} = f'(x_i)$$

- (5.4) formula uler daiteke diferentziak jarraituak direla interpretatuz:

$$f[x_i, x_i] = \lim_{x \rightarrow x_i} f[x, x_i] = \lim_{x \rightarrow x_i} \frac{f(x) - f(x_i)}{x - x_i} = f'(x_i).$$

(ESKEMA)

(ADIBIDEA)

**Oharrak:**

- Interpolazio arrunteko kasuan bezala, Hermite-ren polinomioa balioztatzeke Hörner-en teknikaren bariazioa erabiliko dugu.
- Memorian biltegitatzeko, diferentziak kalkulatu dira azpitik gora, interpolazio arrunteko kasuan legez.

- Hermite-ren polinomio orokortuen problema planteatu daiteke baita,  $x_i$  nodo jakin batean (edo batzuetan)  $f(x_i), f'(x_i), \dots, f^{(k)}(x_i)$  datuak ezagutzen ditugunean. Kasu horretan,  $x_i$  nodo berezi hori(ek) errepikatuko dugu  $(k+1)$  aldiz. Diferentzia zatituen kalkuluan sortuko diren arazoak (5.4) formulaz saihestuko ditugu. Adibidez,

$$f[x_i, x_i, x_i] = \frac{f''(\xi)}{2!}.$$

(ADIBIDEA)

## 5.6 Chebyshev nodoak

Atal honetan interpolaziorako nodoak hautatzeko era optimoa ikusiko dugu errorea minimizatzeko asmoz. Gogoratu errearen adierazpenak

$$|e(x)| = \frac{|f^{(n+1)}(\xi_x)|}{(n+1)!} |\Pi(x)| \quad \text{eta} \quad |e(x)| = \frac{|f^{(2n+2)}(\xi_x)|}{(2n+2)!} \Pi^2(x) \quad (5.5)$$

direla interpolazio arrunt eta Hermite interpolaziorako, hurrenez hurren.  $f$ -ren deribatuak gure kontrolpean ez daudenez (hau da, optimizagarriak ez direnez),  $|\Pi(x)|$  balioa ahalik eta txikiena izaten saiatuko gara. Horretarako, aukeratuko ditugu  $x_0, \dots, x_n$  nodo desberdinak halako moduan non

$$\|\Pi(x)\|_\infty = \max_{x \in [a,b]} |\Pi(x)|$$

balioa ahalik eta txikiena den. Nodo horiek *Chebyshev nodoak* izango dira, definituko ditugun Chebyshev-en polinomioen erroak.

Lehenengoz, aztertuko dugu nodo optimoen hautaketa  $[-1, 1]$  tarterako. Gero, aldagai aldaketa egoki bat eginez, edozein  $[a, b]$  tartetara trasladatuko dugu azterketa.

**5.3 definizioa.** Honako polinomio honi *Chebyshev-en lehen ordenako eta  $n$  mailako polinomioa* deritzo:

$$T_n(t) = \cos[n \cdot \arccos t], \quad \forall t \in [-1, 1]$$

- Nahiz eta iruditu ez, beraz  $T_n(t)$  funtzioa polinomio bat da:

**5.7 proposizioa.** *Chebyshev-en lehen ordenako polinomioek ondorengo formulak egiaztatzen dituzte:*

$$\begin{cases} T_0(t) = 1 \\ T_1(t) = t \\ T_{n+1}(t) = 2t \cdot T_n(t) - T_{n-1}(t), \quad n = 1, 2, \dots \end{cases}$$

*Frogapena:* Lehenengo bi polinomioetarako badaukagu

$$T_0(t) = \cos(0 \cdot \arccos t) = \cos 0 = 1; \quad T_1(t) = \cos(\arccos t) = t.$$

Bestalde,  $\theta = \arccos t$  aldagai aldaketa eginez,

$$\begin{cases} \cos[(n+1) \cdot \theta] = \cos(n \cdot \theta) \cdot \cos \theta - \sin(n \cdot \theta) \cdot \sin(\theta) \\ \cos[(n-1) \cdot \theta] = \cos(n \cdot \theta) \cdot \cos \theta + \sin(n \cdot \theta) \cdot \sin(\theta), \end{cases}$$

Hortaz,

$$\begin{aligned} T_{n+1}(t) + T_{n-1}(t) &= \cos[(n+1) \cdot \theta] + \cos[(n-1) \cdot \theta] \\ &= 2 \cdot \cos(n \cdot \theta) \cdot \cos \theta \\ &= 2t \cdot \cos(n \cdot \theta) = 2t \cdot T_n(t). \end{aligned}$$

Ondorioz,  $T_{n+1}(t) = 2t \cdot T_n(t) - T_{n-1}(t)$ . ♣

- Ikusiriko formula errepikariaren bidez, ondorengo zerrenda lortzen da:

$$\begin{aligned} T_0(t) &= 1 \\ T_1(t) &= t \\ T_2(t) &= 2tT_1(t) - T_0(t) = 2t^2 - 1 \\ T_3(t) &= 2t \cdot T_2(t) - T_1(t) = 4t^3 - 3t \\ T_4(t) &= 2t \cdot T_3(t) - T_2(t) = 8t^4 - 8t^2 + 1 \\ &\dots \end{aligned}$$

- Indukzioz erraz frogatu daiteke  $T_n(t)$ -ren koefiziente nagusia (hau da,  $t^n$ -rekin doan koefizientea)  $2^{n-1}$  dela. Ondorioz,

$$\bar{T}_n(t) = \frac{1}{2^{n-1}} T_n(t)$$

polinomio normalizatua monikoa da.

- Definizioa erabiliz, egiazta daiteke  $T_n(t)$  polinomioaren zeroak (hau da, Chebyshev nodoak  $[-1, 1]$  tartarako) honako hauek direla:

$$t_k = \cos\left(\frac{2k+1}{2n}\pi\right), \quad k = 0, 1, \dots, n-1$$

eta erraz irudikatzen dira: uniformeki bananduta daude 1 erradioko zirkunferentzian, baina proiektatu ondoren, ez dira distantzikideak [IRUDIA].

- Bestalde, ikus daiteke ere  $T_n(t)$  polinomioak maximo eta minimoak hartzen dituztela ondorengo puntuetan:

$$t_k^\bullet = \cos\left(\frac{k\pi}{n}\right), \quad k = 0, 1, \dots, n-1,$$

hartzen dituzten balio maximo eta minimoak  $\pm 1$  izanik, alternatiboki.

**5.8 teorema.** Bedi  $\mathbb{P}_n$ ,  $n$  mailako polinomio moniko guztien multzoa. Orduan,

$$\frac{1}{2^{n-1}} = \|\overline{T}_n\|_\infty \leq \|p\|_\infty, \quad \forall p \in \mathbb{P}_n.$$

(5.5) formulen arabera, eta  $\Pi(x)$  polinomio moniko bat izanik, ondoriozta dezakegu  $T_n(t)$ -ren zeroak nodo egokienak direla interpolazio-errorea gutxitzeko  $[-1, 1]$  tartean. Beraz,  $x_0, \dots, x_n \in [-1, 1]$  edozein nodo badira, eta baldin  $p(x)$  bada interpolatzen dituen polinomioa, orduan

$$\Pi_p(x) = (x - x_0) \dots (x - x_n)$$

polinomioa  $(n + 1)$ -garren mailako polinomio moniko bat da. Bestalde, kontsideratu  $t_0, \dots, t_n$  Chebyshev nodoak ( $T_{n+1}$  polinomioaren zeroak), eta  $q(x)$  nodo horiek erabiliz eraikitzen den interpolazio-polinomioa. Orduan,

$$\Pi_q(x) = (x - t_0) \dots (x - t_n)$$

beste polinomio moniko bat dugu. Aurreko teorema erabiliz, badaukagu

$$\|\overline{T}_{n+1}\|_\infty = \max_{x \in [-1, 1]} |\Pi_q(x)| \leq \max_{x \in [-1, 1]} |\Pi_p(x)|$$

eta horrek esan nahi du Chebyshev nodoak hoberenak direla errorea  $[-1, 1]$  tartean uniformeki optimizatzeko.  $[-1, 1]$  tartetik  $[a, b]$  tartera igarotzeko, honako aldagai aldaketa hau egingo da:

$$x = \frac{b-a}{2} \cdot t + \frac{b+a}{2} \iff t = \frac{2}{b-a} \cdot x - \frac{b+a}{b-a}.$$

Holan,  $[a, b]$  tarterako 1. ordenako eta  $n$  mailako Chebyshev polinomioa dugu:

$$T_n^\bullet(x) = \cos \left[ n \cdot \arccos \left( \frac{2}{b-a} x - \frac{b+a}{b-a} \right) \right]$$

eta ondorengo propietateak beteko ditu:

- $T_n^\bullet(x)$ -ren koefiziente nagusia ondorengo da:

$$2^{n-1} \cdot \left( \frac{2}{b-a} \right)^n.$$

- $T_n^\bullet(t)$  polinomioaren erroak ondorengoak dira:

$$\boxed{x_k = \frac{b-a}{2} \cos \left( \frac{2k+1}{2n} \pi \right) + \frac{b+a}{2}, \quad k = 0, 1, \dots, n-1}$$

Ikusiriko nodo hauek *Chebyshev-en nodoak* dira  $[a, b]$  tarterako.

- $T_n^\bullet(t)$  polinomioak maximo eta minimoak hartzen ditu ondorengo puntuetan:

$$x_k^\bullet = \frac{b-a}{2} \cdot \cos \left( \frac{k\pi}{n} \right) + \frac{b+a}{2} \quad k = 0, 1, \dots, n.$$

## 5.7 Runge efektua

(5.5) formuletatik ondoriozta daiteke  $e(x)$  eta  $\Pi(x)$  funtzioen portaera antzekoa dela. Portaera honi *Runge efektua* deritzo:  $\Pi(x)$  polinomioa oszilagarria da (kontuan hartu bere faktORIZAZIOA).

Nodoak distantzikideak direnean, nabaritzen da tartearen muturretan errorearen oszilazioa tartearen erdiko puntuetan baino handiagoa dela. Gainera, nodoen kopurua haundituz arazo hori larritzen da, egonkortasun eta baldintzatze arazoak direla eta. Bestalde, Chebyshev nodoekin interpolatuz, errorea uniformeki oszilatzen dela ikus daiteke. Ondorioz, tarte osoan lan egin behar badugu, hobe dugu Chebyshev nodoak erabil. Tartearen erdian lan egitekotan, nodo distantzikideak erabil daitezke. (GRAFIKOAK)

## 5.8 Funtzio zatika polinomikoen bidezko interpolazioa

Demagun  $[a, b]$  tarte luzeegi batean lanegiten ari garela. Alde batetik, tarte handia izanik, nodo gehiago hartu behar izango genuke. Bestalde, ikusi dugunez, nodo gehiegi erabiltzen badira, polinomioaren maila handia izanik, errorea handia izan daiteke ere.

Arazo hauek ekiditzeko teknika tartearen partiketa bat egitea da, hau da:

$$[a, b] = \bigcup_{i=0}^{n-1} [x_i, x_{i+1}]$$

tarte nagusia azpitarteetan bananduz.  $[x_i, x_{i+1}]$  azpitarte bakoitzean, kalkulatu da maila baxuko  $p_i(x)$  polinomio interpolatzaile bat, azpitarteko muturrak nodoak izanik (hori lortzeko, beste nodo batzuk hartu behar dira azpitartean). Prozedura honen bidez, lortzen da  $p: [a, b] \rightarrow \mathbb{R}$  funtzio jarraitu bat zeinerako

$$p(x) = p_i(x), \forall x \in [x_i, x_{i+1}], \forall i \in \{0, \dots, n-1\},$$

hau da,  $p$  funtzioa *zatika polinomiala* da. Nahiz eta funtzioa jarraitua izan, bere deribatuak etenak izan daitezke. Grafikoki, erpinak ikusiko dira  $x_i$  puntuetan.

Egonkortasun eta baldintzatze arazoak direla eta, komenigarria izango da nodo gutxi erabiltzea azpitarte bakoitzean. Normalean, *spline*-en teknika erabiltzen da.





# Bibliografía

- [1] J.R. Alonso, A.R. Carrera, *Análisis Numérico, Notas de clase*, Sección de Publicaciones, ETSI Bilbao.
- [2] J.R. Alonso, A.R. Carrera, *Análisis Numérico, Problemas de clase*, Sección de Publicaciones, ETSI Bilbao.
- [3] J.R. Alonso, A.R. Carrera, *Análisis Numérico, Problemas resueltos*, Sección de Publicaciones, ETSI Bilbao.
- [4] J. Stoer, R. Burlich, *Introduction to numerical Analysis*, Springer Verlag, 1980.
- [5] B. Carnahan, H.A. Luther, J.O. Wilkes, *Cálculo Numérico. Métodos, aplicaciones*, Ed. Rueda, 1979.
- [6] R.L. Burden, J.D. Faires, *Análisis Numérico*, Grupo Editorial Iberoamericana, S.A. de C.V., 1985.