

## 2. TÉTELEK LINEÁRIS ALGEBRÁBÓL

Mért adatok kezelésében a mátrixok alkalmazása megkönnyíti a dolgunkat. Ebben a függelékben azokat az ismereteket összegezzük, amelyekre a jegyzet többi részében feltétlenül szükség lesz. A vektorok, mátrixok fogalmát, a velük való műveleteket (szorzás, összeadás, állandóval való szorzás, transzponálás, invertálás, diadikus szorzatok), továbbá az ezekre vonatkozó tételeket ismertnek tételezzük fel. A mátrixok méreteit általában nem jelöljük – kivéve azokat az eseteket, amelyekben ez a képletek megértéséhez és helyes használatához elengedhetetlen. A témakörrel kapcsolatban az Olvasó figyelmébe ajánljuk Rózsa Pál kitűnő könyvét [5].

### 2.1. Sajátértékek, sajátvektorok

Összefoglalónkat a sajátértékekkel és sajátvektorokkal kezdjük. Egy tetszőleges négyzetes  $\mathbf{A}$  mátrix jobb és bal oldali sajátvektorait az

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}, \quad \mathbf{v}^T\mathbf{A} = \lambda\mathbf{v}^T$$

egyenletekkel definiáljuk. A  $\lambda$  sajátértékek a

$$\det(\mathbf{A} - \lambda\mathbf{E}) = P_n(\lambda) = 0 \quad (2.1)$$

karakterisztikus egyenlet megoldásai. Ha az  $\mathbf{A}$  mátrix  $n \times n$ -es,  $P_n(\lambda)$   $n$ -edfokú polinom, amelynek így  $n$  gyöke van. Bizonyítható az

2.1. TÉTEL (HAMILTON-CALEY):

$$P_n(\mathbf{A}) = \mathbf{0}. \quad (2.2)$$

Ha a (2.1) karakterisztikus egyenletnek minden gyöke egyszeres,  $P_n(\lambda)$  a legalacsonyabb fokszámú polinom, amelybe az  $\mathbf{A}$  mátrixot helyettesítve a  $\mathbf{0}$  mátrix adódik. Amikor azonban a polinom gyökei többszörösek, létezhet alacsonyabb fokszámú polinom is, amelyre ugyanez érvényes. Tekintsük az  $(\mathbf{A} - \lambda\mathbf{E})$  mátrix összes  $(n - 1)$ -edrendű aldeterminánsát. Mindegyikük  $\lambda$ -nak legfeljebb  $(n - 1)$ -edfokú polinomja. Ha  $P_n(\lambda)$ -t elosztjuk legnagyobb közös osztójukkal, akkor a  $\Delta(\lambda)$  *minimálpolinom*ot kapjuk. Ez a legalacsonyabb fokszámú polinom, amelybe  $\mathbf{A}$ -t helyettesítve  $\mathbf{0}$ -t kapunk:

$$\Delta(\mathbf{A}) = \mathbf{0}. \quad (2.2a)$$

$\Delta(\lambda)$  általában megegyezik a  $P_n(\lambda)$  karakterisztikus polinommal. A definícióból következik, hogy a minimálpolinom és a karakterisztikus polinom gyökei azonosak, legfeljebb multiplicitásuk lehet különböző.

Az  $\mathbf{A}$  mátrix szerkezetére vonatkozóan sokat elmond a minimálpolinom. Jegyzetünkben gyakran alkalmazzuk a következő tételt:

2.2. TÉTEL. Ha a minimálpolinom minden gyöke egyszeres, az  $\mathbf{A}$  mátrix diagonalizálható:

$$\mathbf{A} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \dots \\ \mathbf{v}_n^T \end{bmatrix} = \mathbf{U} \text{diag}(\lambda) \mathbf{V}^T, \quad (2.3)$$

ahol

$$\mathbf{V}^T \mathbf{U} = \mathbf{U} \mathbf{V}^T = \mathbf{E}. \quad (2.3a)$$

$\mathbf{U}$  oszlopai a jobb oldali,  $\mathbf{V}^T$  sorai pedig a bal oldali sajátvektorok. A fenti képletek érvényességéhez fel kellett tennünk, hogy a sajátvektorok normálása olyan, hogy biortonormált rendszert alkossanak:

$$\mathbf{v}_i^T \mathbf{u}_j = \delta_{ij}.$$

A mátrix (2.3) alakú előállítását *spektrálfelbontásnak* nevezzük. Vannak esetek, amikor nem szükséges a minimálpolinom gyökeit vizsgálni, ugyanis bizonyítható [5] az

2.3. TÉTEL. Ha egy mátrix felcserélhető az adjungáltjával, akkor az diagonalizálható.

Ilyenek a szimmetrikus mátrixok:  $\mathbf{A}^T = \mathbf{A}$ . Ekkor  $\mathbf{V} = \mathbf{U}$ , ahol  $\mathbf{U}$  *unitér mátrix*: inverze megegyezik a transzponáltjával.

A sajátértékekkel kapcsolatban még egy tételre lesz szükségünk, amely szerint felső korlátot kaphatunk a sajátértékek abszolút értékére:

2.4. TÉTEL. Ha az  $\mathbf{A}$  mátrix  $(i, j)$  eleme  $a_{ij}$ , akkor van olyan  $i$ , amelyre mindegyik  $\lambda_k$  sajátérték esetében fennáll, hogy

$$|\lambda_k| \leq \sum_{j=1}^n |a_{ij}|.$$

Írjuk fel a  $\lambda_k$  sajátértékhez tartozó sajátérték-egyenletet ( $i = 1, 2, \dots, n$ ):

$$\sum_{j=1}^n a_{ij} u_{kj} = \lambda_k u_{ki}.$$

Legyen  $i = i'$  az az index, amelyre  $u_{ki}$  abszolút értéke a legnagyobb. Az erre vonatkozó egyenletet osszuk el  $u_{ki}$ -vel, majd vegyük mindkét oldal abszolút értékét:

$$|\lambda_k| = \left| \sum_{j=1}^n a_{i'j} \frac{u_{kj}}{u_{ki'}} \right| \leq \sum_{j=1}^n |a_{i'j}| \cdot \left| \frac{u_{kj}}{u_{ki'}} \right| \leq \sum_{j=1}^n |a_{i'j}|,$$

amint a tétel állítja.

## 2.2. A mátrix rangja

A mátrix fontos tulajdonsága a *rangja*, amelynek több, egymással egyenértékű definíciója van:

2.1. DEFINÍCIÓ. A mátrix rangja  $k$ , ha legfeljebb  $k$  lineárisan független diádra bontható.

Rózsa Pál könyvében [5] található egy algoritmus, amely szerint bármely mátrix egyértelműen felbontható lineárisan független diadikus szorzatok összegére:

$$\mathbf{A} = \sum_{i=1}^k \mathbf{x}_i \mathbf{y}_i^T, \quad (2.4)$$

ahol  $k \leq n$ . A *lineáris függetlenség* azt jelenti, hogy a

$$\sum_{i=1}^k a_i \mathbf{x}_i = \mathbf{0} \quad \text{és} \quad \sum_{i=1}^k b_i \mathbf{y}_i = \mathbf{0}$$

egyenlőségek csak úgy állhatnak fenn, hogy az  $a_i$  és  $b_i$  együtthatók eltűnnek minden  $i$ -re. Nos, az  $\mathbf{A}$  mátrix rangja definíció szerint  $k$ .

2.2. DEFINÍCIÓ. A mátrix rangja a 0-tól különböző  $\lambda_i$  sajátértékek száma.

A (2.3) felbontásból következik, hogy

$$\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{u}_i \mathbf{v}_i^T. \quad (2.5)$$

Ha csak a zérustól különböző sajátértékeket vesszük, akkor egy (2.4) alakú diadikus felbontást kapunk. Ha abban a diádok száma  $k$ , bizonyítható, hogy megegyezik a (2.4)-ben szereplő  $k$ -val.

2.3. DEFINÍCIÓ. A mátrix rangja  $k$ , ha van el nem tűnő  $k$ -adrendű aldeterminánsa, de minden magasabb rendű aldetermináns zérus.

Annak bizonyításától is eltekintünk, hogy ez az előbbiekkal ekvivalens. Megjegyezzük, hogy ezt a definíciót ritkán használjuk a jegyzetben. A rangszámmal kapcsolatban gyakran alkalmazzuk a következő tételt:

2.5. TÉTEL. A rang nem növelhető szorzás révén.

Azt kell belátnunk, hogy bármely két mátrixra

$$\text{rang}(\mathbf{AB}) \leq \text{rang}(\mathbf{A}) \quad \text{és} \quad \text{rang}(\mathbf{AB}) \leq \text{rang}(\mathbf{B}). \quad (2.6)$$

Szorozzuk be jobbról a (2.4) egyenlet mindkét oldalát  $\mathbf{B}$ -vel:

$$\mathbf{AB} = \sum_{i=1}^k \mathbf{x}_i \mathbf{y}_i^T \mathbf{B} = \sum_{i=1}^k \mathbf{x}_i \mathbf{z}_i^T,$$

ahol  $\mathbf{z}_i^T = \mathbf{y}_i^T \mathbf{B}$  ( $i = 1, 2, \dots, k$ ). Ha ezek a vektorok lineárisan függetlenek, akkor az  $\mathbf{AB}$  szorzatmátrix rangja maradt  $k$ . Ha azonban lineárisan függők, akkor az utolsó összegben a lineárisan független diádok száma  $k$ -nál kisebb, így  $\mathbf{AB}$  rangja is kisebb, mint  $\mathbf{A}$ -é. Ezt állítja a tétel. Hasonlóan láthatjuk be a (2.6) képletek közül a másodikat is.

A mátrixok rangjával kapcsolatba hozhatók a kvadratikus formák, amelyeket a következőképpen definiálunk:

2.4. DEFINÍCIÓ. Legyen  $\mathbf{A}$  szimmetrikus, valós elemű mátrix. Az

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j \quad (2.7a)$$

összeget *kvadratikus formának* nevezzük.

Alkalmazzuk a (2.3) alatti felbontást:

$$\mathbf{x}^T \mathbf{A} \mathbf{x} = \mathbf{x}^T \mathbf{U}^T \mathbf{A} \mathbf{U} \mathbf{x} = \sum_{i=1}^n \lambda_i z_i^2, \quad (2.7b)$$

ahol

$$\mathbf{z} = \mathbf{U} \mathbf{x}.$$

Tudjuk, hogy szimmetrikus, valós elemű mátrixok sajátértékei valósak.<sup>1</sup> Ennek figyelembevételével értelmezhetők a következő definíciók:

2.5. DEFINÍCIÓ. A szimmetrikus, valós elemű  $\mathbf{A}$  mátrixot *pozitív definitnek* nevezzük, ha minden sajátértéke pozitív ( $\lambda_i > 0$ ). Ebben az esetben a (2.7) szerinti kvadratikus alak minden nemzérus  $\mathbf{x}$  vektor esetében pozitív.

2.6. DEFINÍCIÓ. A szimmetrikus, valós elemű  $\mathbf{A}$  mátrixot *pozitív szemidefinitnek* nevezzük, ha minden sajátértéke nemnegatív ( $\lambda_i \geq 0$ ). Ebben az esetben a (2.7) szerinti kvadratikus alak minden nemzérus  $\mathbf{x}$  vektor esetében pozitív vagy zérus.

2.7. DEFINÍCIÓ. A szimmetrikus, valós elemű  $\mathbf{A}$  mátrixot *indefinitnek* nevezzük, ha sajátértékei különböző előjelűek. Ebben az esetben a (2.7) szerinti kvadratikus alak egyaránt felvehet pozitív, negatív és zérus értéket.

---

<sup>1</sup> Az idézett tétel pontosabban úgy hangzik, hogy hermitikus mátrixok sajátértékei valósak [5]. Hermitikus mátrix: megegyezik komplex konjugáltjának a transzponáltjával.

Analóg módon definiálható a *negatív definit* és *negatív szemidefinit* mátrix fogalma is. A 2.2. DEFINÍCIÓból következik, hogy a pozitív és negatív *definit* mátrixok rangja megegyezik a mátrix rendjével.

Pozitív definit és pozitív szemidefinit mátrixok esetében általánosíthatjuk a valós számok körében ismert gyökvonás műveletét. Erre vonatkozik az

2.6. TÉTEL. Tetszőleges szimmetrikus, valós elemű, pozitív definit vagy pozitív szemidefinit  $\mathbf{A}$  mátrixhoz található olyan valós elemű  $\mathbf{H}$  mátrix, amelyre  $\mathbf{A} = \mathbf{H}^T \mathbf{H}$ .

Bizonyítás gyanánt felírunk egy ilyen  $\mathbf{H}$  mátrixot:

$$\mathbf{H}^T = [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] \begin{bmatrix} \sqrt{\lambda_1} & 0 & 0 \\ 0 & \sqrt{\lambda_2} & 0 \\ 0 & 0 & \sqrt{\lambda_n} \end{bmatrix} = \mathbf{U} \text{diag}(\sqrt{\lambda})$$

(2.3) alapján egyszerűen beláthatjuk, hogy ez megfelel a tétel kívánalmainak. Megjegyezzük, hogy a  $\mathbf{H}$  mátrix előállításához nem feltétlenül szükséges az  $\mathbf{A}$  mátrix (2.3) szerinti spektrálfelbontását elvégezni, mert vannak gyorsabb algoritmusok is. (Ilyen a *Housholder-algoritmus*, lásd [5].)

### 2.3. Mátrixok invertálása

#### A probléma felvetése

Mátrixokkal kapcsolatban két művelet igényel különösen gondosan kidolgozott algoritmusokat: a (2.3) szerinti spektrálfelbontás és az invertálás. Mérések kiértékelésekor az előbbi konkrét végrehajtására nincs nagyon szükség. Elméleti levezetésekben persze annál többször alkalmazzuk a 2.2. TÉTELT. Bizonyos mátrixok inverzét azonban minden esetben numerikusan ki kell számítanunk. Erre vonatkozóan Gauss óta kidolgozott algoritmus létezik: *elimináció előre, visszahelyettesítés hátra*. Elérhetőek ezt megvalósító szubrutinok, amelyek feladatukat több-kevesebb sikerrel meg is oldják. Mivel a mátrixok invertálása alapvető szerepet játszik a mérések kiértékelésében, a problémával részletesen foglalkozunk.

Előljáróban megjegyezzük még, hogy az alább tárgyalt algoritmust csak alacsony rendű mátrixok esetében célszerű alkalmazni. Ennek csak egyik oka, hogy a kivonási jegyvesztések problémája a mátrix rendjének ( $n$ ) növekedésével egyre súlyosabbá válik. A másik ok az, hogy a Gauss-féle algoritmus  $n^3$ -nel arányos számú szorzást és osztást igényel, vagyis a szükséges számítási idő a mátrix rendjével rohamosan nő. Ezért nagy mátrixok esetében olyan iterációs algoritmusok kerülnek előtérbe, amelyekben a szorzások és osztások száma csak  $n^2$ -tel arányosan nő. Nehéz általánosságban megadni, hol van a két megközelítés közötti határ. Körülbelül a  $100 \times 100$ -as mátrixok jelentik azt a határt, amely felett fontolóra lehet venni az iterációs eljárásokat. Speciális szerkezetű mátrixok esetében a határ jelentősen kitolható a 2.4. alfejezetben tárgyalt hipermátrixok segítségével.

Jóllehet az említett algoritmus általában ismert, egy példa kapcsán felidézzük a lépéseit. Tekintsük a következő egyszerű egyenletrendszer megoldását:

$$\begin{aligned} 1,01x + 2y + 3z &= 1 \\ 4x + 5y + 6z &= 0 \\ 7x + 8y + 9z &= 0 \end{aligned} \quad (2.8a)$$

Az előre való elimináció első lépéseként az első egyenletet végigosztjuk  $x$  együtthatójával, majd az így kapott egyenletet végigszorozzuk 4-gyel és 7-tel, majd az eredményt kivonjuk a második, illetve a harmadik egyenletből:

$$\begin{aligned} x + 1,9802y + 2,9703z &= 0,9901 \\ -2,9208y - 5,8812z &= -3,9604 \\ -5,8614y - 11,7921z &= -6,9307 \end{aligned} \quad (2.8b)$$

A későbbi tanulságok levonása érdekében az első osztás eredményét négy tizedesjegyre kerekítettük, és ezt követően végig a kerekített számokkal dolgoztunk, majd az újabb eredményeket szintén négy tizedesjegyre kerekítettük. A későbbi lépésekben is így járunk el. Ezzel illusztráljuk a számítógépek véges számítási pontosságát és annak numerikus következményeit. Az  $x$  ismeretlenek azt az együtthatóját, amellyel először osztottunk, *pivotelemnek* nevezzük. Látható, hogy a kapott egyenletrendszer két utolsó egyenletéből kiküszöböltük (elimináltuk) az  $x$  változót. Ezt követően  $y$ -t küszöböljük ki az utolsó két egyenletből. Pivotelemünk most  $-2,9208$ , vagyis az  $y$  ismeretlenek a második egyenletben szereplő együtthatója. Ezzel végigosztjuk a második egyenletet, majd az eredményt végigszorozzuk az  $y$  ismeretlenek a harmadik egyenletben szereplő együtthatójával,  $-5,8614$ -gyel, és az eredményt kivonjuk a harmadik egyenletből:

$$\begin{aligned} x + 1,9802y + 2,9703z &= 0,9901 \\ y + 2,0136z &= 1,3559 \\ 0,0104z &= 1,0168 \end{aligned} \quad (2.8c)$$

Ezzel az algoritmus “elimináció előre” részét végrehajtottuk. Ha több ismeretlenünk és egyenletünk lenne, akkor ezt tovább folytatva küszöbölnénk ki az ismeretleneket. Végeredményben olyan egyenletrendszert kapunk, amiben az utolsó egyenlet csak egy változót tartalmaz, majd visszafelé haladva mindig eggyel több ismeretlen jelenik meg. Mielőtt továbbmennénk, felhívjuk a figyelmet arra, hogy a harmadik egyenletben a  $z$  együtthatójának a számításakor végzett kivonásban az első három értékes tizedesjegy kiesett (elveszett).

Az algoritmus második része “visszahelyettesítés hátra”. Az utolsó egyenletből (itt: a harmadikból) kifejezzük  $z$ -t, majd visszafelé haladva közvetlenül kiszámítjuk az egyes ismeretleneket:

$$\begin{aligned} x &= 0,9901 - 1,9802 \cdot (-195,512) - 2,9703 \cdot 97,769 = 97,740 \\ y &= 1,3559 - 2,0136 \cdot 97,769 = -195,512 \\ z &= 1,0168 / 0,0104 = 97,769 \end{aligned} \quad (2.8d)$$

Itt ugyan öt értékes számjegyet írtunk le, de ezek közül legfeljebb az első három lehet értelmes, hiszen a (2.8c)-hez vezető lépésben a harmadik egyenletben  $z$  együtthatóját csak ilyen pontossággal tudtuk megkapni. Az eredeti egyenletrendszer nem nehéz pontosan is megoldani:

$$\begin{aligned}x &= 100 \\y &= -200 \\z &= 100\end{aligned}\tag{2.8e}$$

A (2.8d) szerinti eredmény a vártnál is rosszabb: a véges számítási pontosság okozta hiba miatt már a második számjegy is megbízhatatlan.

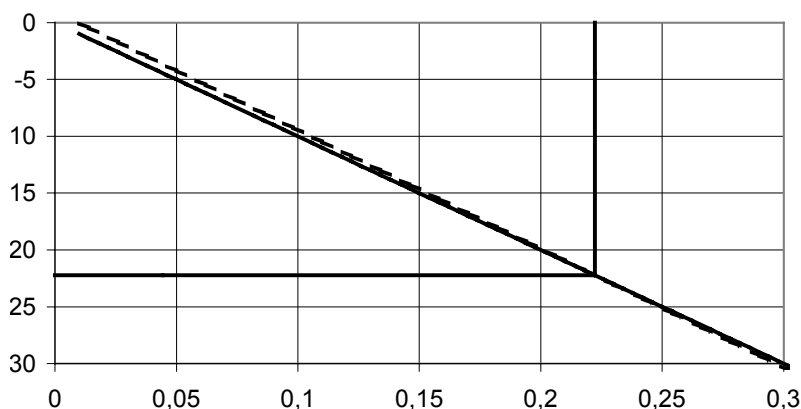
A megoldást nyilván nagyobb pontossággal szeretnénk megkapni. A bemutatott példából – reméljük – világos, hogy a probléma gyökere a *kivonási jegyvesztés*, ugyanis ez az a művelet, amelyben értékes számjegyek veszhetnek el, amikor az egymásból kivonandó számok közelítőleg azonosak. Rögtön felmerül két kérdés: Mikor lép fel ez a kellemetlen jelenség? Ha fellép, lehet-e ellene valamit tenni? Az alábbiakban ezekre keressük a választ.

Mielőtt rátérnénk a válaszokra, megjegyezzük, hogy a (2.8a) egyenletrendszer megoldása megadja a mátrix inverzének első oszlopát. A másik két oszlopot úgy kapjuk, hogy a jobb oldalra az eddigi  $[1, 0, 0]$  vektor helyett a  $[0, 1, 0]$ , illetve  $[0, 0, 1]$  vektorokat írjuk. Egy mátrix invertálására tehát ugyanúgy alkalmazhatjuk az “elimináció előre, visszahelyettesítés hátra” algoritmusát. Az is látszik továbbá, hogy a fellépő numerikus problémák elméleti vizsgálatát elég az egyenletrendszerek megoldására korlátozni.

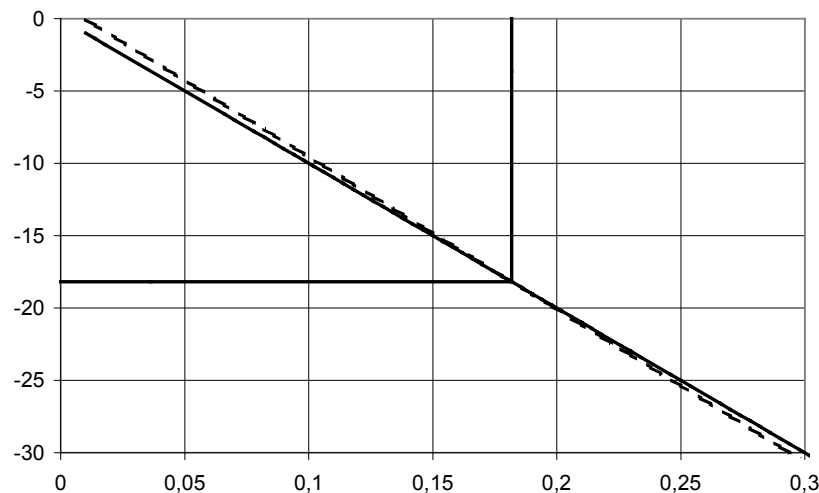
E rész befejezéseként kimondjuk a következő tételt:

**2.7. TÉTEL.** A pivotelemek szorzata megadja a mátrix inverzét.

A tétel bizonyítása az ismertett algoritmusból következik, és az Olvasóra bízunk. Útmutatás: az előre való helyettesítés felső háromszög mátrixszá transformálja a mátrixot, amelynek determinánsa a főátlóban levő elemek szorzata, vagyis 1. A transzformáció során azonban beszorzunk a pivotelemeket tartalmazó diagonális mátrix inverzével. A sorok néhányszorosának a kivonása nem változtatja meg a determinánst.



2.1a. ábra. A (2.9) egyenletrendszer grafikus megoldása



2.1b. ábra. A módosított (2.9) egyenletrendszer grafikus megoldása

### Geometriai szemléltetés

A (2.8) képletekben vizsgált harmadrendű mátrix rajzban nehezen szemléltethető. Ezért a probléma lényegét egy kétismeretlenes egyenletrendszer segítségével szemléltetjük:

$$\begin{array}{l}
 104,5x + y = 1 \\
 x + 0,01y = 0
 \end{array}
 \text{ pontos megoldása: }
 \begin{array}{l}
 x = 1/(104,5 - 100) = 0,2222 \\
 y = -100x = -22,22
 \end{array}
 \quad (2.9)$$

Ha az egyenletrendszert alkotó egyenleteket az  $(x, y)$  síkon ábrázoljuk, két egyenest kapunk. Metszéspontjuk koordinátái adják az egyenletrendszer megoldását. A 2.1a. ábrán látható, hogy a két egyenes majdnem párhuzamos egymással. Ilyen esetben lépnek fel a kivonási jegyveszteségek, mint (2.9) megoldásában:  $104,5 - 100 = 4,5$ . A gyakorlatban ez arra vezet, hogy a megoldás nagyon érzékeny a mátrix elemeire és a jobb oldalon álló ismert vektorra. Változtassuk meg például az első egyenletben  $x$  együtthatóját 104,5-ről 105,5-re! Ekkor a 2.1b. ábrán látható képet kapjuk, amely szerint a megoldás:  $x = 0,1818$  és  $y = -18,18$ . Az együttható 1%-nyi megváltozása a megoldásban 18% eltérést eredményezett. Az ilyen tulajdonságú egyenletrendszerekről azt mondjuk, hogy *rosszul kondicionáltak*.

### Rosszul kondicionált mátrixok

A (2.8a) egyenletrendszer esetében a megoldást az  $(x, y, z)$  térben felvett síkok metszéspontja adja. Nos, ott a jegyveszteség azzal függ össze, hogy a síkok közel párhuzamosak egymással. A kondicionáltság fokának számszerű jellemzésére a következő geometriai megfontolást alkalmazzuk. A (2.9) egyenletrendszer mátrixának az oszlopvektorai:

$$\mathbf{a}_1 = \begin{bmatrix} 104,5 \\ 1 \end{bmatrix} \quad \text{és} \quad \mathbf{a}_2 = \begin{bmatrix} 1 \\ 0,01 \end{bmatrix}$$



Az  $(x, y)$  síkon ezek kifeszítenek egy parallelogrammát. Területe a mátrix determinánsának abszolút értéke:  $|\det \mathbf{A}| = 0,045$ . Ez azért kis szám, mert a két oszlopvektor közel párhuzamos egymással. Ha merőlegesek lennének, a parallelogramma téglalap lenne, amelynek a területe

$$|\mathbf{a}_1| \cdot |\mathbf{a}_2| \approx 104,5 \cdot 1 = 104,5.$$

A két terület hányadosa a kondicionáltság számszerű mértékének tekinthető. Általában

$$C_1(\mathbf{A}) = \frac{|\det \mathbf{A}|}{|\mathbf{a}_1| \cdot |\mathbf{a}_2| \cdot \dots \cdot |\mathbf{a}_n|}, \quad (2.10)$$

ahol a nevezőben az  $n \times n$ -es  $\mathbf{A}$  mátrix oszlopvektorainak a hossza szerepel. A (2.8a) és a (2.9) mátrixokra ez rendre  $4,3 \cdot 10^{-4}$  és  $3,4 \cdot 10^{-5}$ .  $\det \mathbf{A}$  kiszámításában támaszkodhatunk a 2.7. TÉTELRE.

A kondicionáltságra egy másik mérőszámot is levezethetünk. Ehhez szükségünk van a mátrix valamilyen normájára. Először a vektorok normáját kell definiálnunk, majd ahhoz a következő módon kapcsolhatunk a mátrixok számára is normát. Legyen  $\mathbf{x}$  tetszőleges (nemzérus) vektor, és tekintsük az  $\mathbf{Ax}$  vektorokat. Nos, a mátrix normája a legkisebb olyan  $M$  szám, amelyre fennáll az

$$\|\mathbf{Ax}\| \leq M\|\mathbf{x}\|$$

egyenlőtlenség. Ebből következik, hogy minden  $\mathbf{x}$  vektorra fennáll:

$$\|\mathbf{Ax}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|. \quad (2.11)$$

A vektorok számára a továbbiakban az  $L_2$  normát használjuk:

$$\|\mathbf{x}\| = \sqrt{\mathbf{x}^T \mathbf{x}} = \sqrt{\sum_{i=1}^n x_i^2}. \quad (2.12)$$

Belátjuk, hogy ehhez a vektornormához tartozó mátrixnorma

$$\|\mathbf{A}\| = \lambda_1, \quad (2.13)$$

ahol  $\lambda_1^2$  az  $\mathbf{A}^T \mathbf{A}$  mátrix legnagyobb sajátértéke. Feltesszük, hogy az  $\mathbf{A}$  mátrix szimmetrikus, és így (2.3) szerint faktorizálható.<sup>2</sup> Ha bevezetjük a

$$\mathbf{z} = \mathbf{V}^T \mathbf{x}$$

jelölést, akkor

$$\mathbf{Ax} = \mathbf{U} \text{diag}(\lambda) \mathbf{V}^T \mathbf{x} = \sum_{i=1}^n \lambda_i z_i \mathbf{u}_i,$$

amelynek a normája

<sup>2</sup> A mérések kiértékelésénél fellépő mátrixokra ez a feltevés mindig teljesül.

$$\|\mathbf{Ax}\|^2 = \mathbf{x}^T \mathbf{A}^T \mathbf{Ax} = \sum_{i=1}^n \lambda_i^2 z_i^2.$$

Itt kihasználtuk, hogy az  $\mathbf{u}_i$  sajátvektorok 1-re vannak normálva. Tegyük fel, hogy a sajátértékeket csökkenő sorrendben indexeltük, vagyis  $\lambda_1$  a legnagyobb abszolút értékű. Ekkor írhatjuk:

$$\|\mathbf{Ax}\|^2 \leq \lambda_1^2 \sum_{i=1}^n z_i^2 = \lambda_1^2 \mathbf{z}^T \mathbf{z} = \lambda_1^2 \mathbf{x}^T \mathbf{x} = \lambda_1^2 \|\mathbf{x}\|^2.$$

Ha  $\mathbf{x} = \mathbf{u}_1$ , akkor itt egyenlőség áll fenn, amivel állításunkat igazoltuk.

Könnyen beláthatók a mátrixnormára vonatkozóan a következő összefüggések. Két mátrix szorzatának a normája nem lehet nagyobb, mint a tényezők normájának a szorzata:

$$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|. \quad (2.14a)$$

Fennállnak továbbá a háromszög-egyenlőtlenségek:

$$\|\mathbf{A}\| - \|\mathbf{B}\| \leq \|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|. \quad (2.14b)$$

A fentiek segítségével vizsgáljuk az

$$\mathbf{Ax} = \mathbf{b} \quad (2.15)$$

egyenletrendszer megoldásait. Tekintsük először a  $\mathbf{b}$  vektor perturbációit. Ha  $\mathbf{b}$ -t megváltoztatjuk  $\Delta\mathbf{b}$ -vel, akkor a megoldás megváltozása

$$\Delta\mathbf{x} = \mathbf{A}^{-1} \Delta\mathbf{b}.$$

Vegyük mindkét oldal normáját, és alkalmazzuk a fenti egyenlőtlenségeket:

$$\begin{aligned} \|\Delta\mathbf{x}\| &\leq \|\mathbf{A}^{-1}\| \cdot \|\Delta\mathbf{b}\| = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{b}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{Ax}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} \leq \\ &\leq \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\| \cdot \|\mathbf{x}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}, \end{aligned}$$

amit a következő alakban is írhatunk:

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\| \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|} = C_2(\mathbf{A}) \frac{\|\Delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

Itt bevezettük a következő jelölést:

$$C_2(\mathbf{A}) = \|\mathbf{A}^{-1}\| \cdot \|\mathbf{A}\|. \quad (2.16)$$

Ez a mennyiség ugyanúgy jellemzi az  $\mathbf{A}$  mátrix kondicionáltságát, mint a (2.10) szerinti  $C_1$ , hiszen megadja, hogy a  $\mathbf{b}$  vektor relatív megváltozása hányszorosan felnagyítva jelentkezik a megoldásban.

Nem számoljuk végig, de meg lehet mutatni, hogy ugyanez a  $C_2$  mérvadó magának az  $\mathbf{A}$  mátrixnak a perturbációi szempontjából is. Ha a mátrixot  $\Delta\mathbf{A}$ -val perturbáljuk, akkor a megoldás relatív megváltozását a

$$\frac{\|\Delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{C_2(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}}{1 - C_2(\mathbf{A}) \frac{\|\Delta\mathbf{A}\|}{\|\mathbf{A}\|}}$$

képlet jellemzi. Mindenképpen fennáll tehát, hogy a (2.15) egyenletrendszer akkor rosszul kondicionált, amikor  $C_2$  nagy. Mivel az inverz mátrix sajátértékei az eredeti mátrix reciprokai, az  $\|\mathbf{A}^{-1}\|$  mátrixnorma nem más, mint  $\mathbf{A}$  legkisebb abszolút értékű sajátértékének a reciproka. Ezzel tehát

$$C_2(\mathbf{A}) = \left| \frac{\lambda_1}{\lambda_n} \right|. \quad (2.16a)$$

A fenti két példában szereplő mátrixokra ennek értéke:  $2,4 \cdot 10^5$  (2.9)-re és  $1,06 \cdot 10^4$  (2.8a)-ra. Mindkettő meglehetősen nagy érték.

### Algoritmus

Miután számszerűen jellemeztük a mátrixok kondicionáltságát, nézzük, hogyan lehet a helyzetet javítani. (2.16a)-ból látszik, hogy nagyon kedvezőtlen, ha az invertálandó mátrixnak nagyságrendileg eltérő sajátértékei vannak. A mérések kiértékelésekor könnyen keletkezhet ilyen mátrix, ha a mért mennyiségek egységeit kedvezőtlenül választjuk meg. A kísérleti fizikusoktól azonban nem várható el, hogy ilyesmire ügyeljenek. Az ebből esetleg keletkező problémákat a mérést kiértékelő programok keretében kell elintézni. Az invertálandó mátrixot célszerű transzformálni: jobbról és balról beszorozzuk egy-egy diagonális mátrixszal úgy, hogy a transzformált mátrix sor- és oszlopvektorainak a normája közelítőleg azonos, mondjuk, 1 legyen. Ezzel általában javul a kondicionáltság. Matematikailag tehát arról van szó, hogy az  $\mathbf{A}$  mátrix helyett az

$$\mathbf{A}' = \mathbf{R}\mathbf{A}\mathbf{S} \quad (2.17a)$$

mátrixot invertáljuk, majd a kapott eredményből az

$$\mathbf{A}^{-1} = \mathbf{S}\mathbf{A}'^{-1}\mathbf{R} \quad (2.17b)$$

képlettel kapjuk meg a keresett inverzet. Példaképpen vegyük a (2.9)-ben szereplő mátrixot. Könnyen belátható, hogy az

$$\mathbf{S} = \mathbf{R} = \begin{bmatrix} 0,1 & 0 \\ 0 & 10 \end{bmatrix}$$

választás megfelelő, mert ez az

$$\mathbf{A}' = \begin{bmatrix} 1,045 & 1 \\ 1 & 1 \end{bmatrix}$$

mátrixra vezet, amelyre (2.16a) alapján  $C_2(\mathbf{A}') = 90,9$  adódik, ami lényegesen kedvezőbb, mint korábban ( $2,4 \cdot 10^5$ ). A (2.17) transzformációk is járnak kerekítéssel, ami természetesen újabb hibaforrás. Ennek kiküszöbölésére egyes programok a transzformáló diagonális mátrixokban csak olyan elemeket engednek meg, amelyek 2 vagy 10 egész számú hatványai. Ilyesmire azonban csak végső esetben van szükség.

A következő numerikus fogás a pivotelem megfelelő megválasztása. A tapasztalat azt mutatja, hogy a kivonási jegyvesztések akkor a legkisebbek, amikor az előre való kiküszöbölés mindegyik lépésében az éppen kiküszöbölt változó együtthatói közül az abban az egyenletben szereplőt választjuk pivotelemnek, amelyiknek legnagyobb az abszolút értéke.

Amint az algoritmus halad előre, az így kiválasztott pivotelem abszolút értéke egyre kisebb. Ha a mátrix szinguláris vagy olyan rosszul kondicionált, hogy az adott gépi pontosság mellett az inverz mátrix elemeinek egyetlen számjegye sem vehető komolyan, akkor az algoritmust le kell állítani. Ha végtelen pontossággal számolnánk, a szinguláris mátrixok esetében a pivotelem nullává válna a  $(k+1)$ -edik lépésben, ahol  $k$  a mátrix rangja. Véges pontosságú számítások esetében ez sohasem következik be. Legfeljebb arról lehet szó, hogy a pivotelem abszolút értéke egy bizonyos toleranciaérték ( $TOL$ ) alá csökken, ami annak a jele, hogy a mátrix szinguláris, vagy legalábbis az adott gépi pontosság mellett annak tekintendő. Tetszőleges mátrixok esetében nem lehet ilyen határt megadni. Az invertálandó mátrix normálása ebből az okból is tanácsos. Normált mátrixok esetében a következő határok szabhatók:  $TOL = 10^{-6}$  a szokásos négybájtos számábrázolás (az ún. *szimpla precízió*), viszont  $TOL = 10^{-13}$  a nyolcbájtos számábrázolás (az ún. *dupla precízió*) esetén.

### Utóiteráció

Vannak esetek, amelyekben a fenti fogások sem segítenek. Ilyen például a (2.8a)-ban szereplő mátrix: az oszlopvektorok normája közel azonos, és a pivotelem optimális megválasztása sem javítja lényegesen az invertálás pontosságát. Az utolsó lehetőség ilyenkor az *utóiteráció*, amellyel szinte "reménytelen" eseteket is meg lehet menteni. Lehet alkalmazni akár az inverz, akár az  $\mathbf{Ax} = \mathbf{b}$  egyenlet közelítő megoldásának a javítására. Tegyük fel, hogy a fenti numerikus eljárásokkal kaptunk egy közelítő  $\mathbf{B} \approx \mathbf{A}^{-1}$  inverz mátrixot. Az alábbi séma bal oldalán az egyenletrendszerre, a jobb oldaliban pedig az inverzre vonatkozó algoritmust mutatjuk:

$$\begin{array}{l|l} \mathbf{x}_0 = \mathbf{Bb}, & \mathbf{X}_0 = \mathbf{B} \\ \mathbf{b}_1 = \mathbf{b} - \mathbf{Ax}_0 = (\mathbf{E} - \mathbf{AB})\mathbf{b} = \mathbf{Zb}, & \mathbf{B}_1 = \mathbf{E} - \mathbf{AX}_0 = \mathbf{E} - \mathbf{AB} = \mathbf{Z}, \\ \mathbf{x}_1 = \mathbf{Bb}_1 = \mathbf{BZb}, & \mathbf{X}_1 = \mathbf{BB}_1 = \mathbf{BZ}, \\ \dots & \dots \\ \mathbf{x}_k = \mathbf{Bb}_k = \mathbf{BZ}^k \mathbf{b}, & \mathbf{X}_k = \mathbf{BB}_k = \mathbf{BZ}^k, \end{array}$$

$$\begin{array}{l|l}
\mathbf{b}_{k+1} = \mathbf{b}_k - \mathbf{A}\mathbf{x}_k = & \mathbf{B}_{k+1} = \mathbf{B}_k - \mathbf{A}\mathbf{X}_k = \\
= (\mathbf{E} - \mathbf{A}\mathbf{B})\mathbf{b}_k = \mathbf{Z}^{k+1}\mathbf{b}, & = \mathbf{Z}^k - \mathbf{A}\mathbf{B}\mathbf{Z}^k = \mathbf{Z}^{k+1}, \\
\hline
\sum_{k=0}^{\infty} \mathbf{x}_k = \sum_{k=0}^{\infty} \mathbf{B}\mathbf{Z}^k \mathbf{b} = \mathbf{B} \sum_{k=0}^{\infty} \mathbf{Z}^k \mathbf{b} = & \sum_{k=0}^{\infty} \mathbf{X}_k = \mathbf{B} \sum_{k=0}^{\infty} \mathbf{Z}^k = \mathbf{B}(\mathbf{E} - \mathbf{Z})^{-1} = \\
= \mathbf{B}(\mathbf{E} - \mathbf{Z})^{-1} \mathbf{b} = \mathbf{A}^{-1} \mathbf{b} = \mathbf{x}. & = \mathbf{B}(\mathbf{A}\mathbf{B})^{-1} = \mathbf{B}\mathbf{B}^{-1} \mathbf{A}^{-1} = \mathbf{A}^{-1}.
\end{array}$$

Ezek a sorok konvergálnak, ha  $\mathbf{Z}$ -nek nincs 1-nél nagyobb sajátértéke. Erre vonatkozóan jó becslést kaphatunk a 2.4. TÉTEL alapján. Ha tehát  $\mathbf{B}$  legalább 1 jegyre jó inverz, akkor  $\mathbf{Z}$ -nek nincs 0,1-nél nagyobb eleme, így az utóiteráció 10-nél kevesebb lépésben konvergál (10 tizedesjegyre).

Példaként tekintsük a (2.8a) szerinti mátrixot. A (2.8) képletekben alkalmazott algoritmussal, négy tizedes jegyre való kerekítéssel a következő közelítő inverzet kapjuk:

$$\mathbf{B} = \begin{bmatrix} 97,74 & -195,57 & 97,80 \\ -195,51 & 388,22 & -193,62 \\ 97,77 & -192,97 & 96,15 \end{bmatrix}$$

Az iteráció sebességét a

$$\mathbf{Z} = \mathbf{E} - \mathbf{A}\mathbf{B} = \begin{bmatrix} -0,0074 & -0,0043 & 0,012 \\ -0,03 & 0 & 0 \\ -0,03 & -0,04 & 0,01 \end{bmatrix}$$

mátrix határozza meg. A 2.4. TÉTEL szerint ennek legnagyobb abszolút értékű sajátértékére (a harmadik sor alapján) a következő felső becslést kapjuk:  $|\lambda| \leq 0,03 + 0,04 + 0,01 = 0,08$ , ami azt jelenti, hogy minden iterációs lépésben a korrekció – nagyjából – egy nagyságrenddel csökken. Nézzük ezt részleteiben:

$$\mathbf{X}_1 = \mathbf{B}\mathbf{Z} = \begin{bmatrix} 2,21 & -4,33 & 2,15 \\ -4,39 & 8,59 & -4,28 \\ 2,18 & -4,27 & 2,13 \end{bmatrix}$$

$$\mathbf{X}_2 = \mathbf{B}\mathbf{Z}^2 = \begin{bmatrix} 0,049 & -0,096 & 0,048 \\ -0,097 & 0,190 & -0,095 \\ 0,048 & -0,095 & 0,047 \end{bmatrix}$$

$$\mathbf{X}_3 = \mathbf{B}\mathbf{Z}^3 = \begin{bmatrix} 0,001 & -0,002 & 0,001 \\ -0,002 & 0,004 & -0,003 \\ -0,001 & -0,001 & 0,001 \end{bmatrix}$$

Látható, hogy már az első iteráció ( $\mathbf{X}_1$ ) elég lenne, ha csak négy értékes számjegyre keresnénk az inverzet. A második iteráció ( $\mathbf{X}_2$ ) azonban a második tizedesjegyre helyesen adja meg az inverzet:

$$\mathbf{A}^{-1} \cong \mathbf{B} + \mathbf{X}_1 + \mathbf{X}_2 = \begin{bmatrix} 100,00 & -200,00 & 100,00 \\ -200,00 & 397,00 & -198,00 \\ 100,00 & -197,34 & 98,33 \end{bmatrix}$$

Ezzel a pontossággal  $\mathbf{X}_3$  és a további iterációk elhanyagolhatók. Beszorzással ellenőrizhetjük, hogy ez – az adott pontossággal – valóban  $\mathbf{A}$  inverze.

## 2.4. Hipermatrixok

Előfordul, hogy a kezelendő mátrixokat célszerű blokkokra bontani, és az így keletkező almatrixokat egy mátrix elemeinek tekinteni. Így értelmezzük a *hipermatrix*okat. Elméletüket részletesen kidolgozza Rózsa Pál könyve [5]. Jegyzetünkben csak a  $2 \times 2$ -es hipermatrixokra lesz szükségünk:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$$

Közvetlen behelyettesítéssel ellenőrizhetjük, hogy a szimmetrikus hipermatrix inverze visszavezethető alacsonyabb rendű mátrixok invertálására. A későbbi hivatkozás érdekében ezt egy tétel formájában fogalmazzuk meg:

2.8. TÉTEL. Szimmetrikus hipermatrix inverzét a következő képlet adja meg:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix}^{-1} = \begin{bmatrix} (\mathbf{A} - \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C})^{-1} & -(\mathbf{A} - \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{B}^{-1} \\ -\mathbf{B}^{-1} \mathbf{C} (\mathbf{A} - \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C})^{-1} & \mathbf{B}^{-1} + \mathbf{B}^{-1} \mathbf{C} (\mathbf{A} - \mathbf{C}^T \mathbf{B}^{-1} \mathbf{C})^{-1} \mathbf{C}^T \mathbf{B}^{-1} \end{bmatrix}$$

Ez a képlet hasznos, ha a  $\mathbf{B}$  blokk inverzét könnyű kiszámítani: például akkor, amikor  $\mathbf{B}$  diagonális mátrix, vagy amikor  $\mathbf{B}$  alacsony rendű mátrix. Ezekben az esetekben egyszerűsíthető a mátrix négyzetgyöke is. Erre vonatkozik az

2.9. TÉTEL. Szimmetrikus hipermatrix négyzetgyökét a következő képlet adja meg:

$$\begin{bmatrix} \mathbf{A} & \mathbf{C}^T \\ \mathbf{C} & \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_A & \mathbf{0} \\ \mathbf{H}_C & \mathbf{H}_B \end{bmatrix}^T \begin{bmatrix} \mathbf{H}_A & \mathbf{0} \\ \mathbf{H}_C & \mathbf{H}_B \end{bmatrix}.$$

Az egyes blokkok kielégítik az

$$\mathbf{A} = \mathbf{H}_A^T \mathbf{H}_A + \mathbf{H}_C^T \mathbf{H}_C,$$

$$\mathbf{B} = \mathbf{H}_B^T \mathbf{H}_B,$$

$$\mathbf{C} = \mathbf{H}_B^T \mathbf{H}_C.$$

Az algoritmust  $\mathbf{B}$  faktorizációjával kezdjük. Ezt követően a harmadik egyenletből kapjuk a  $\mathbf{H}_C$  blokkot, amelyet az első egyenletbe helyettesítve számítjuk ki  $\mathbf{H}_A$ -t. Ez az eljárás főleg akkor kifizetődő, amikor  $\mathbf{B}$  diagonális, és  $\mathbf{A}$  mérete nem túlságosan nagy.