DETERMINÁNSOK

Determináns kiszámítási módjai,
tulajdonságai és alkalmazásai.

1. Determináns

A determináns fogalmának kiépítése többféleképpen is megtörténhet. Van, aki a determináns egy külön objektumnak tekinti. Mi jobban szeretjük a determinánst úgy interpretálni, hogy ez egy leképezés, ami négyzetes mátrixokhoz rendel számot. Azt hogy hogy, a következő alfejezetekben fogjuk definiálni.

1.1. Sarrus-szabály

Egy 1 darab számból álló mátrix determinánsa maga a mátrixot alkotó szám. Egy (2×2)-es determináns kiszámítására maga a Sarrus-szabály a definíció:

$$\begin{aligned} \begin{vmatrix} a & b \\ c & d \end{vmatrix} &= (\text{főátló elemeinek szorzata}) - (\text{mellékátló elemeinek szorzata}) \\ &= ad - bc. \end{aligned}$$

Felhívjuk a <u>következő</u> rész veszélyére a figyelmet. A Sarrus-szabály csak (3×3)-as méretig működik, nagyobb mátrixra NEM használható.

A (3×3)-as mátrix determinánsa hasonlóan számítható. Itt nem csak a főátlóval, és a mellékátlóval kell számolni, hanem a velük párhuzamos „átlókkal” is. Segítségképpen a determináns után odaírhatjuk az első két oszlopát, hogy jobban lássuk a párhuzamosságot.

$$\left|\begin{array}{ccccc} a & & b & & c \\ & \searrow & & \searrow & \\ d & & e & & f \\ & & & \searrow & \\ g & & h & & i \end{array}\right| \begin{array}{cccc} & a & & b \\ \searrow & & & \\ & d & & e \\ \searrow & & \searrow & \\ & g & & h \end{array}, \qquad \left|\begin{array}{ccccc} a & & b & & c \\ & & & \swarrow & \\ d & & e & & f \\ & \swarrow & & \swarrow & \\ g & & h & & i \end{array}\right| \begin{array}{cccc} & a & & b \\ \swarrow & & \swarrow & \\ & d & & e \\ \swarrow & & & \\ & g & & h \end{array}$$

A determináns a következőképp áll össze. A főátlóban és a vele párhuzamos átlókban lévő elemek szorzatát adjuk össze, és ebből vonjuk ki a mellékátlóban, és a vele párhuzamos átlóban lévő elemek szorzatát. Tehát a következőt kell csinálni:

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh.$$

Nézzük meg ezt egy konkrét példán keresztül.

Példa.

$$\begin{vmatrix} 3 & 2 & -4 \\ -2 & -5 & 1 \\ -8 & 2 & 1 \end{vmatrix} = (3\cdot(-5)\cdot 1) + (2\cdot 1\cdot(-8)) + ((-4)\cdot(-2)\cdot 2)$$
$$- ((-4)\cdot(-5)\cdot(-8)) - (2\cdot(-2)\cdot 1) - (3\cdot 1\cdot 2)$$
$$= -15 - 16 + 16 + 160 + 4 - 6 = 143$$

1. *Megjegyzés.* Maga a számolási algoritmus nem bonyolult, de ennél a módszernél viszonylag nagy az elszámolás veszélye.

Felhívjuk az <u>előző</u> rész veszélyére a figyelmet. A Sarrus-szabály csak (3×3)-as méretig működik, nagyobb mátrixra NEM használható.

1.2. Sor/oszlop szerinti kifejtés (Kifejtési tétel)

Most mutatunk egy olyan módszert, amivel tetszőleges méretű mátrix determinánsa kiszámolható. (Vigyázzunk, nagy méretű mátrixok esetén nagyon megnő a módszer számolásigénye.)

Sor, illetve oszlop szerinti kifejtésnél gondolnunk kell a mátrixhoz tartozó „sakktáblára”, ami előjeleket tartalmaz felváltva, a bal felső sarok mindig „+”, és onnantól váltakozik az előjel jobbra és lefelé, mint egy sakktábla színei:

+	−	+	…
−	+	−	…
+	−	+	…
⋮	⋮	⋮	⋱

Kiválasztjuk a determináns tetszőleges sorát vagy oszlopát, ami szerint a kifejtést el akarjuk végezni. Legyen ez először például az **első oszlop**. A determináns értéke a következőképpen adódik. A kiválasztott sor, vagy oszlop elemeit egyesével megszorozzuk a sakktáblában neki megfelelő előjellel, majd megszorozzuk annak a maradék determinánsnak az értékével, amit úgy kapunk, hogy az eredeti determinánsból töröljük az elem sorát és oszlopát. Az így kapott értékeket összeadva kapjuk meg a determináns értékét. A példán talán jobban látszik, hogy hogyan kell csinálni.

Példa. A determinánsban a piros előjelek a sakktábla megfelelő elemeit jelölik.

$$\begin{vmatrix} 3^{+} & 2 & -4 \\ -2^{-} & -5 & 1 \\ -8^{+} & 2 & 1 \end{vmatrix} = +3\cdot\begin{vmatrix} -5 & 1 \\ 2 & 1 \end{vmatrix} - (-2)\cdot\begin{vmatrix} 2 & -4 \\ 2 & 1 \end{vmatrix} + (-8)\cdot\begin{vmatrix} 2 & -4 \\ -5 & 1 \end{vmatrix} \tag{1}$$

A módszer lényege, hogy egy $(n \times n)$ méretű determinánst vissza tudunk vezetni $(n-1) \times (n-1)$-es determinánsokra. Folytassuk az (1) egyenlőséget, mivel a (2×2)-es determinánsok értéke már könnyen számolható:

$$\begin{aligned}(1) &= 3 \cdot ((-5) \cdot 1 - 1 \cdot 2) + 2 \cdot (2 \cdot 1 - (-4) \cdot 2) - 8 \cdot (2 \cdot 1 - (-4) \cdot (-5)) \\ &= 3 \cdot (-7) + 2 \cdot 10 - 8 \cdot (-18) = -21 + 20 + 144 = 143.\end{aligned}$$

Ellenőrzésképpen végezzük el a determináns kifejtését a **második sora** szerint is:

$$\begin{aligned}\begin{vmatrix} 3 & 2 & -4 \\ -2^{-} & -5^{+} & 1^{-} \\ -8 & 2 & 1 \end{vmatrix} &= -(-2) \cdot \begin{vmatrix} 2 & -4 \\ 2 & 1 \end{vmatrix} + (-5) \cdot \begin{vmatrix} 3 & -4 \\ -8 & 1 \end{vmatrix} - 1 \cdot \begin{vmatrix} 3 & 2 \\ -8 & 2 \end{vmatrix} \\ &= 2 \cdot (2 \cdot 1 - (-4) \cdot 2) - 5 \cdot (3 \cdot 1 - (-4) \cdot (-8)) - 1 \cdot (3 \cdot 2 - 2 \cdot (-8)) \\ &= 2 \cdot 10 - 5 \cdot (-29) - 1 \cdot 22 = 20 + 145 - 22 = 143.\end{aligned}$$

2. *Megjegyzés.* Ez a módszer tetszőleges méretű determinánsokra is működik, szemben a Sarrus-szabállyal, ami csak (2×2)-es és (3×3)-as determinánsokra alkalmazható.

3. *Megjegyzés.* Ha van a mátrix elemei között nulla, akkor érdemes lehet olyan sort, vagy oszlopot választani a kifejtéshez, amiben hemzsegnek a nullák. Ugyanis a kifejtésnél a nullához tartozó kisebb determináns 0-val szorzódna, így fel sem fontos tüntetni.

Példa.

$$\begin{vmatrix} 0 & 2 & 0 & 3 \\ 1 & 2 & 0 & -5 \\ 10 & 6 & -3 & 4 \\ 0 & 1 & 0 & 2 \end{vmatrix} = -3 \begin{vmatrix} 0 & 2 & 3 \\ 1 & 2 & -5 \\ 0 & 1 & 2 \end{vmatrix} = (-1)(-3) \begin{vmatrix} 2 & 3 \\ 1 & 2 \end{vmatrix} = (-3)(-1)(4-3) = 3$$

A fenti determinánst a harmadik oszlopa szerinti kifejtéssel határoztuk meg. Így egy 3×3-as determinánst kaptunk, amit pedig az első oszlopa szerinti kifejtéssel kaptunk meg.

4. *Megjegyzés.* Dolgozatokban érdemes a kifejtéses módszert alkalmazni, mert elszámolási hiba esetén még esetleg lehet részpontot adni, míg a Sarrus-szabálynál ez nehezebben oldható meg.

1.3. Elemi átalakítások (Gauss-elimináció)

Ez a módszer ugyanúgy alkalmazható tetszőleges méretű mátrixokra, és sokkal gyorsabb is, viszont nem olyan egyszerű algoritmus szerint működik, mint az előző kettő. Egyébként nagyon fontos algoritmus elméleti és gyakorlati szempontból is. A kurzuson is elő fog még kerülni több kontextusban.

5. Tétel (Determinánsokra vonatkozó alapvető tulajdonságok.)**.**

1. *Egy determináns előjelet vált, ha két sorát megcseréljük.*
2. *Ha egy determináns valamelyik sora nulla, akkor a determináns értéke nulla.*
3. *Egy determináns értéke nulla, ha van két azonos sora.*
4. *Ha a determináns egy sorában minden elemet ugyanazzal a nemnulla konstanssal megszorzunk, vagy elosztunk, akkor a determináns értéke is ezzel a konstanssal szorzódik, vagy osztódik.*
5. *Egy determináns értéke nulla, ha az egyik sora egy másik sor valamely konstans-szorosa.*
6. ***A determináns értéke nem változik, ha valamelyik sorhoz hozzáadjuk egy másik sor konstans-szorosát.***
7. *Dualitási elv: az* $1-6$. *állításokban a „sor” szó kicserélhető az „oszlop” szóra.*

A félkövérrel kiemelt tulajdonság ismételt használatával gyorsan ki tudjuk számítani a determinánst, mert meg tudjuk növelni a determinánsban szereplő nullák számát. A felsorolás többi eleme is hasznos lehet, de az alkalmazásuk nem szükségszerű.

Példa.

$$\begin{vmatrix} 3 & 2 & -4 \\ -2 & -5 & 1 \\ -8 & 2 & 1 \end{vmatrix} \overset{(1)}{=} \begin{vmatrix} 3 & 2 & -4 \\ 6 & -7 & 0 \\ -8 & 2 & 1 \end{vmatrix} \overset{(2)}{=} \begin{vmatrix} -29 & 10 & 0 \\ 6 & -7 & 0 \\ -8 & 2 & 1 \end{vmatrix}$$
$$\overset{(3)}{=} 1 \cdot \begin{vmatrix} -29 & 10 \\ 6 & -7 \end{vmatrix} \overset{(4)}{=} (-7)\cdot(-29) - 6\cdot 10 = 143.$$

- (1): A 2. sorhoz hozzáadtam a 3. sor (-1)-szeresét, azaz alkalmaztam a Tétel 6. állítását.
- (2): Az 1. sorhoz hozzáadtam a 3. sor 4-szeresét, azaz alkalmaztam a Tétel 6. állítását.
- (3): Kifejtettem a determináns a 3. oszlopa szerint. (A sok nulla miatt valójában csak egy kisebb determinánst kell kiszámolni.)
- (4): Sarrus-szabállyal megkaptam a végső eredményt.

1.4. Érdekesség

Mindkét említett általános algoritmus könnyen programozható, és egy 100×100-as mátrix determináns kiszámítását nyilván nem kézzel fogjuk kiszámítani. Azonban számítógép használata esetén minden számítási algoritmusnál meg kell vizsgálni az alábbi két tulajdonságot.

1. Milyen gyors az algoritmus?
2. Mennyire pontos az algoritmus?

Az utóbbit itt most nem boncolgatjuk, viszont az első kérdésre a fenti algoritmusok tekintetében meglepő eredmények adhatók.

Egy $(n\times n)$-es mátrix determinánsának kiszámítása a kifejtéses módszerrel $\mathcal{O}(n!)$ időigénnyel tehető meg. Ha Gauss-eliminációt használunk, akkor bizonyítható, hogy az időigény csak $\mathcal{O}(n^3)$. Röviden rávilágítanánk arra, hogy mennyivel jelent ez nagyobb gyorsaságot. Tegyük fel, hogy egy 5 GHz-es processzorú számítógéppel számolunk, ami azt jelenti, hogy 5 milliárd műveletet tud elvégezni másodpercenként. A könnyítés kedvéért a továbbiakban csak az ordo utáni függvényekkel számolunk. (A kapott eredményeket valami konstanssal meg kellene szorozni, a nagyságrendeken ez nem változtatna.)

Ha $n = 3$, akkor a mátrix determinánsának kiszámítása kifejtéses módszerrel $0,12 \cdot 10^{-8}$ másodpercig tart, míg Gauss-eliminációval $0,54 \cdot 10^{-8}$ másodpercig. Itt még a kifejtéses módszer a gyorsabb. Ha $n = 10$, akkor az első módszerrel a számítás $0,00072576$ másodpercig tart, Gauss-eliminációval $0,0000002$ másodpercig tart. Itt az első módszer már több mint 3000-szer lassabb, de gyakorlatilag ez még elhanyagolható, mert a végső idő itt is kicsi.

Nézzük az $n = 15$ esetet. Gauss-eliminációval az időigény $0,000000675$ másodperc, míg kifejtéses módszerrel $4,358914560$ perc. Ez már eléggé érzékelhető és zavaró különbség. Ha $n = 20$, akkor a Gauss-eliminációnak még mindig jóval másodperc alatti idő szükséges, $0,0000016$ másodperc, míg a kifejtéses módszernek $15,64366003$ évre lenne szüksége. Nem lenne túl hatékony ezt kivárni, és még nagyon messze vagyunk a (100×100)-as mérettől.

Végül ugorjunk egy „hatalmasat”, legyen $n = 50$. A Gauss-elimináció még mindig hatékony, időigénye $0,000025$ másodperc. A kifejtéses módszerrel már komoly problémába ütköznénk, ugyanis $0.1955638709 \cdot 10^{48}$ évre lenne szüksége. A Nap várható hátralévő élettartamát 5-10 milliárd évre becsülik, tehát a Nap már nem élné meg az eredményt. Ha jól tudjuk, akkor ez a számolási időigény már a világegyetem várható életkoránál is nagyobb szám. Ha valaki kíváncsi rá, hogy a Gauss-elimináció mekkora n esetén megy 1 másodperc fölé, az könnyen kiszámolhatja egy egyszerű egyenletmegoldással. A fenti adatokat foglalja össze az alábbi táblázat.

Méret	Gauss-elimináció	Rekurzív kifejtés
3×3	$0,54 \cdot 10^{-8}$ mp	$0,12 \cdot 10^{-8}$ mp
10×10	$0,0000002$ mp	$0,00072576$ mp
15×15	$0,000000675$ mp	$4,358914560$ perc
20×20	$0,0000016$ mp	$15,64366003$ év
50×50	$0,000025$ mp	$0.1955638709 \cdot 10^{48}$ év

Más kérdés, hogy melyik módszer mennyire stabil numerikusan, ebbe most nem megyünk bele, de ez is érdekes kérdés.

A vázolt probléma egyáltalán nem csak elméleti, mert bizonyos területeken valóban több százszor több százas méretű mátrixokkal kell számolni, és ráadásul ezek a mátrixok több tíz tizedesjegy pontosságú tizedes törteket tartalmaznak. Hasonló kérdésekkel találkozhattok a Közelítő és szimbolikus számítások kurzuson.

1.5. Alkalmazások

- Kalkulus: Jacobi-determináns
- Paralelogramma területének, paralelepipedon térfogatának kiszámítása.

- Egyenletrendszerek megoldása Cramer-szabállyal.
- Adott pontokon átmenő sík, egyenes, kör egyenlete is felírható determinánssal.
- Adott egy gráf. Van-e benne kör? Mennyi a feszítőfáinak száma? A kérdések a megfelelő mátrixok determinánsa alapján megoldható. (Informatikai párhuzam: van-e holtpont az erőforrások között? A választ lásd az Operációs rendszerek című kurzuson.)
- Adott egy páros gráf, van-e benne teljes párosítás? Egy speciális determinánssal ez is könnyen eldönthető.

1.6. Kiegészítés

- Wolfram Alpha / Wolfram Mathematica: `Det[3,2,-4,-2,-5,1,-8,2,1]`
- Maple: `LinearAlgebra[Determinant](Matrix([[3,2,-4],[-2,-5,1],[-8,2,1]]))`
- Matlab: `det([3 2 -4;-2 -5 1;-8 2 1])`
- Matek.hu