DOI: 10.15476/ELTE.2021.151

Eötvös Loránd Tudományegyetem
Informatikai Kar
Komputeralgebra Tanszék

Algebrai számokkal kapcsolatos algoritmusok és számítások

Doktori értekezés

Uray Marcell János

Témavezető:
Burcsi Péter
(egyetemi docens)

Doktori iskola: Informatika Doktori Iskola

Doktori iskola vezetője: Dr. Csuhaj Varjú Erzsébet
(tanszékvezető egyetemi tanár)

Doktori program: Numerikus és Szimbolikus Számítások

Programvezető: Dr. Weisz Ferenc (egyetemi tanár)

Budapest, 2021

Tartalom

Tartalom **3**

1. Bevezetés **5**

1.1. A dolgozat felépítése 5

1.2. Köszönetnyilvánítás 7

2. Futási idők algebrai számtestben **9**

2.1. Bevezetés 9

2.1.1. Algebrai számok áttekintése 9

2.1.2. A fejezet jelölései 11

2.2. Alapvető műveletek 12

2.2.1. Méretnövekedés a műveletek során 13

2.2.2. A műveletek futási ideje 17

2.2.3. Műveletek összefoglalása 22

2.3. Bareiss-algoritmus 23

2.4. LLL-algoritmus 25

2.4.1. Tulajdonságok $\mathbb{R}^n$-ben 27

2.4.2. Változók mérete $\mathbb{Z}[\theta]^n$-ben 29

2.4.3. Az algoritmus futási ideje $\mathbb{Z}[\theta]^n$-ben 33

2.4.4. Megjegyzések az LLL-eredményről 36

3. Expanzív polinomok **37**

3.1. Bevezetés 37

3.1.1. Fogalmak és jelölések 37

3.1.2. Korábbi eredmények 38

3.2. Speciális determinánsok 40

3.2.1. Az expanzivitás D-feltételei 41
3.2.2. Az eldöntési algoritmus 45
3.2.3. Általánosítás: D-polinomok 47
3.3. Alsó korlátok az expanzivitási hézagra 51
3.3.1. Liouville-típusú korlátok 52
3.3.2. A determinásokból adódó korlátok 54
3.3.3. Korlátok összehasonlítása 57
3.4. Alig expanzív polinomcsalád 58
3.4.1. Motzkin-háromszög és inverze 59
3.4.2. A polinomcsalád és tulajdonságai 60
3.4.3. A bizonyítás befejezése 64
3.5. Általánosított számrendszerek eldöntési algoritmusának futási ideje . . . 66
3.5.1. Definíciók és előzmények 66
3.5.2. A véges halmaz mérete 68
3.5.3. A futási idő 71
3.6. További kutatási lehetőségek 72

Összefoglalás **73**
Summary in English 74

Publikációk **75**

Irodalomjegyzék **76**

1. Bevezetés

Ebben a doktori értekezésben algebrai számokkal kapcsolatos különféle algoritmusokról, valamint ezekhez kapcsolódó elméleti kérdésekről lesz szó. Algebrai számnak nevezzük az olyan számot, amely egy egész együtthatós polinom gyöke. Például ilyen a $\sqrt{2}$, mert az $x^2 - 2$ polinom gyöke.

A vizsgált algoritmusok fontos jellemzője, hogy nem *numerikusan*, azaz közelítésekkel, hanem *egzaktul* (szimbolikusan) számolnak. A $\sqrt{2}$ algebrai szám így sosem 1,4142, hanem mindig pontosan a $\sqrt{2}$ matematikai objektum. Ezzel elkerüljük a numerikus számolások jellemző problémáját, a kerekítési hibákat, amelyek hosszú számítás során összeadódhatnak, és bizonyos esetekben akár teljesen rossz eredményhez is vezethetnek. Az egzakt algoritmusoknak másfajta problémáik vannak: általában lassabbak, mint a numerikus megfelelőjük, és jellemző rájuk az együtthatórobbanás, ami azt jelenti, hogy az egzakt ábrázoláshoz szükséges egész számok (együtthatók) menet közben túl gyorsan, akár exponenciális mértékben növekedhetnek, ami kihat a futási időre is. Ennek elkerülése és a futási idő kordában tartása (polinomiális idő garantálása) fontos szempont az egzakt algoritmusok esetén, és a dolgozatban ez a kérdés többször elő fog kerülni.

Ez a munka nagyrészt a szerző saját publikációin alapul: [2, 3, 4, 5], ezek közül [5] más szerzőkkel közös munka. A szerzőnek további publikációja [6], de annak témája jelentősen eltér a többitől, ezért a dolgozat koherenciájának megőrzése érdekében az ottani eredmények nem kerültek bele.

1.1. A dolgozat felépítése

A disszertáció két nagy témakörre bontható. A 2. fejezetben algebrai számokon végzett műveletek és algoritmusok futási idejét vizsgáljuk. Bár algebrai számokról és a kapcsolódó algoritmusokról számos könyv és cikk szól (pl. [14, 18, 31]), ezek

hatékonyságáról már kevesebb szó esik a szakirodalomban. Ennek precíz elemzése nem könnyű feladat, mert figyelembe kell venni az említett együtthatónövekedést is. A 2.2. szakaszban – egy-két korábbi eredményt is felhasználva – kiszámítjuk a fontosabb műveletek hatékonyságát algebrai számtestekben, köztük olyanokét is, mint a valós algebrai számok összehasonlítása (pl. $\alpha < \beta$) és az egészre kerekítés. Ezekre támaszkodva megvizsgáljuk két algoritmusnak, a Bareiss-algoritmusnak (2.3. szakasz) és az LLL-algoritmusnak (2.4. szakasz) a futási idejét, ha algebrai számokkal végezzük. Az előbbi a Gauss-elimináció módosítása, amely egész számok esetén célzott egzakt osztásokkal elkerüli az együtthatórobbanást. Az LLL-algoritmus egy rácsredukciós algoritmus, amelynek a viselkedése ismert, ha numerikusan számolunk, vagy ha egész számokkal számolunk egzaktul. Mindkét algoritmust kiterjesztjük egzakt algebrai számokra, és bebizonyítjuk, hogy ebben az esetben is polinomiális a futási idejük.

A 3. fejezetben speciális algebrai számokról lesz szó: olyan egész együtthatós polinomok gyökeiről, melyeknek minden gyöke 1-nél nagyobb abszolút értékű (azaz az egységkörön kívül vannak, ha a komplex számsíkon ábrázoljuk őket). Az ilyen polinomokat expanzív polinomoknak nevezzük. Először azzal a kérdéssel foglalkozunk, hogy hogyan tudjuk eldönteni egy polinomról, hogy expanzív-e (anélkül, hogy ki kelljen számítani a gyökeit). Erre a szakirodalomban vannak módszerek, de azok legtöbbször nem kifejezetten egész együtthatós polinomokkal foglakoznak, és nem figyelnek az együtthatórobbanásra. A 3.2. szakaszban bemutatunk egy új módszert az expanzivitás eldöntésére, amely bizonyos determinánsokat értékel ki, és bebizonyítjuk róla, hogy polinomiális idő alatt elvégezhető. Látni fogjuk, hogy ez a módszer nemcsak az említett problémát oldja meg, hanem más ehhez kapcsolódó elméleti kérdésekben is segítségünkre lesz. Ilyen például az, hogy egész együtthatós expanzív polinomok gyökei mennyire lehetnek közel az egységkörhöz. A 3.3. szakaszban – a determinánsos módszerünk segítségével – alsó korlátot számítunk ki erre (a fokszám és az együtthatók méretének függvényében). A 3.4. szakaszban, annak igazolására, hogy a kiszámított alsó korlátunk elég éles, konstruálunk egy expanzív polinomcsaládot, amelyről – szintén a determinánsok segítségével – bebizonyítjuk, hogy vannak az egységkörhöz nagyon közeli gyökei. Végül mutatunk egy alkalmazást az említett alsó korlátra: segítségével meg tudjuk becsülni egy általánosított számrendszerekkel kapcsolatos algoritmus futási idejét (3.5. szakasz).

1.2. Köszönetnyilvánítás

Köszönet illeti

- Burcsi Pétert
  - a hosszú évek kitartó közös munkájáért,
  - az értékes szakmai és nem szakmai támogatásáért,
  - a cikkek és a dolgozat megírásában nyújtott alapos és nem elkapkodott segítségéért,
  - a kutatási témák felvetéséért,
  - a konstruktív hozzáállásáért,
  - a szabad és kellemes légkör megteremtéséért,
  - rugalmasságáért, türelméért és humoráért;
- Kovács Attilát
  - az általánosított számrendszerek bemutatásáért
  - és félúton a helyes irányba terelésért;
- a cikkek névtelen bírálóit értékes javaslataikért,
  - külön kiemelve azt, aki a Liouville-egyenlőtlenséget javasolta;
- mindazokat, akik a matematika oktatásával új szemléletet tudtak adni,
  - külön kiemelve Uray Lászlót és Hubert Györgynét;
- az Eötvös Loránd Tudományegyetemet a színvonalas és szabad környezetért, amelyet remélhetőleg sokáig meg tud őrizni;
- valamint mindazon szabad szoftverek fejlesztőit, amelyek nélkül ez a dolgozat nem jöhetett volna létre: LaTeX, SageMath, GNU eszközök, Linux stb.

2. Futási idők algebrai számtestben

2.1. Bevezetés

Ebben a fejezetben algebrai számokkal való számolásokat fogunk hatékonysági szempontból elemezni. A fejezet nagyrészt a szerző [2] publikációján alapul.

2.1.1. Algebrai számok áttekintése

Az alábbiakban összegyűjtöttük az algebrai számokkal kapcsolatos legfontosabb fogalmakat és ábrázolási módokat. Részletesebb és általánosabb tárgyalásért lásd pl. Cohen könyvét: [14, 4.1-4.2], valamint a szerző korábbi munkáját, [1]-et, amelyben implementáció is található.

2.1.1. Definíció. Az $\alpha \in \mathbb{C}$ számot *algebrai szám*nak nevezzük, ha létezik olyan egész együtthatós $f \neq 0$ polinom, amelynek α gyöke, azaz $f(\alpha) = 0$.

2.1.2. Definíció. Az algebrai számok halmazát $\mathbb{A}$-val jelöljük.

Algebrai számokat számítógépen többféleképpen lehet ábrázolni. A legkézenfekvőbb, ha a definícióból indulunk ki, és az f polinomot tároljuk. Több ilyen polinom is létezik, ezek közül célszerű a következőt választani:

2.1.3. Definíció. Egy $\alpha \in \mathbb{A}$ esetén azt a legalacsonyabb fokú nemnulla egész polinomot, amelynek α gyöke, az együtthatóinak legnagyobb közös osztója 1 (azaz *primitív polinom*), és a főegyütthatója pozitív, az α *minimálpolinomjá*nak nevezzük.

2.1.4. Definíció. Egy α algebrai szám *fokszáma* a minimálpolinomjának foka.

2.1.5. Lemma. *A minimálpolinom irreducibilis, és adott $\alpha \in \mathbb{A}$ esetén egyértelmű.*

Mivel egy minimálpolinomnak több gyöke is lehet (pontosan annyi, amekkora a fokszáma), ezért a gyökök megkülönböztetésére további adatok szükségesek, például egy intervallum.

A minimálpolinomos ábrázolásban nem magától értetődő kérdés a műveletvégzés. Ehhez szükség van a rezultáns fogalmára [14, 119-121. o.]:

2.1.6. Definíció. Legyen $f(x) = f_n x^n + \ldots + f_1 x + f_0$ és $g(x) = g_m x^m + \ldots + g_1 x + g_0$ két egész együtthatós polinom, és legyenek f gyökei $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{C}$, g gyökei pedig $\beta_1, \beta_2, \ldots, \beta_m \in \mathbb{C}$ (multiplicitással számolva). Ekkor a két polinom x szerinti *rezultánsa* a következő szám:

$$\mathrm{res}_x(f, g) := f_n^m g_m^n \prod_{i=1}^{n} \prod_{j=1}^{m} (\alpha_i - \beta_j).$$

2.1.7. Lemma. *A rezultáns felírható a polinomok együtthatóiból összeadás, kivonás és szorzás segítségével, kövekezésképp* $\mathrm{res}_x(f, g) \in \mathbb{Z}$.

A rezultáns segítségével műveleteket tudunk végezni algebrai számok között a minimálpolinomos ábrázolásban [14, 159. o.]:

2.1.8. Tétel. *Ha α egy n-edfokú, β egy m-edfokú algebrai szám, és minimálpolinomjuk $f(x)$, ill. $g(x)$, akkor*

- *$\alpha + \beta$ gyöke a $h(x) = \mathrm{res}_y(f(x - y),\, g(y))$ polinomnak,*
- *$\alpha - \beta$ gyöke a $h(x) = \mathrm{res}_y(f(x + y),\, g(y))$ polinomnak,*
- *$\alpha\beta$ gyöke a $h(x) = \mathrm{res}_y(y^n f(x/y),\, g(y))$ polinomnak,*
- *α/β gyöke a $h(x) = \mathrm{res}_y(f(xy),\, g(y))$ polinomnak,*

és ezen $h(x)$ polinomok fokszáma: nm.

A rezultánst kiszámíthatjuk például a szubrezultáns algoritmussal polinomiális időben [14, 3.3.7. Alg.]. Az eredményként kapott polinom nem feltétlenül irreducibilis, ezért a minimálpolinom meghatározásához szükség lehet polinomfaktorizációra is. A fokszámok (faktorizáció után is) legrosszabb esetben összeszorzódnak, így hosszabb számítás során akár exponenciálisan is növekedhetnek, ezért ez az ábrázolás a gyakorlatban nem alkalmas bonyolultabb algoritmusok elvégzésére.

A másik ábrázolási lehetőség, hogy egy adott feladathoz ideiglenesen leszűkítjük $\mathbb{A}$-t, és egy algebrai számtestben dolgozunk:

2.1.9. Definíció. A K testet *algebrai számtest*nek (röviden: számtest) nevezzük, ha $\mathbb{Q} \subseteq K \subset \mathbb{A}$, és $\mathbb{Q}$ feletti vektortérként véges dimenziós. Ezt a dimenziót nevezzük K *fokszámának.*

Egy számtestet meg tudunk adni egyetlen algebrai szám segítségével:

2.1.10. Definíció. Egy K számtest *primitív eleme* olyan $\theta \in K$, amelyre $\mathbb{Q}(\theta) = K$.

2.1.11. Tétel (primitív elem tétel)**.** *Minden K algebrai számtestnek van primitív eleme.*

A θ primitív elem segítségével minden $\alpha \in K$ elemet egyértelműen fel tudunk írni a következő formában:

$$\alpha = a_0 + a_1\theta + a_2\theta^2 + \ldots + a_{m-1}\theta^{m-1} \qquad (a_i \in \mathbb{Q}), \tag{2.1.1}$$

ahol m a számtest és a primitív elem fokszáma.

Ebben az ábrázolásban az összeadás, a kivonás és a racionális számmal való szorzás és osztás triviális vektorművelet. Az általános szorzás és osztás, mint látni fogjuk, ennél bonyolultabb, de még mindig sokkal gyorsabb, mint minimálpolinomos ábrázolásban rezultánsokkal.

A primitív elemes ábrázolást általánosíthatjuk úgy, hogy nem egyetlen algebrai számot veszünk, hanem m darabot: legyen $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\} \subset K$, amely K-ban, mint $\mathbb{Q}$ feletti vektortérben bázist alkot, és akkor minden $\alpha \in K$ egyértelműen felírható a következő alakban:

$$\alpha = a_1\omega_1 + a_2\omega_2 + \ldots + a_m\omega_m \qquad (a_i \in \mathbb{Q}). \tag{2.1.2}$$

A primitív elemes felírás ennek speciális esete: $\Omega = \{1, \theta, \theta^2, \ldots, \theta^{m-1}\}$.

2.1.2. A fejezet jelölései

Az alábbiakban rögzítjük a 2. fejezetben használt feltevéseket és jelöléseket. Ezeket a továbbiakban külön említés nélkül adottnak tekintjük.

Legyen $K \subset \mathbb{C}$ egy m-edfokú algebrai számtest. A K-beli algebrai számokat a $\theta \in K$ primitív elemmel fogjuk ábrázolni. Jelöljük θ minimálpolinomját f-fel:

$$f(x) = f_m x^m + f_{m-1}x^{m-1} + \ldots + f_1 x + f_0 \qquad (f_i \in \mathbb{Z}).$$

Feltesszük, hogy $f_m = 1$ (azaz θ *algebrai egész*). Könnyen belátható, hogy minden algebrai számtestben van ilyen primitív elem is.

A számtest elemeit fel tudjuk írni θ segítségével a (2.1.1) alakban. A racionális együtthatókkal azonban körülményes számolni, ezért inkább közös nevezőre hozzuk őket, azaz a következő felírást használjuk:

$$\alpha = \frac{b_0 + b_1\theta + b_2\theta^2 + \ldots + b_{m-1}\theta^{m-1}}{d} \qquad (b_i \in \mathbb{Z},\ d \in \mathbb{Z}^+),$$

ahol $b_0, b_1, \ldots, b_{m-1}$ és d legnagyobb közös osztója 1. Sőt, mivel a nevező kezelése az elmélet szempontjából triviális, és, mint látni fogjuk, számos algoritmusnál előre felszorozhatunk vele, ezért a továbbiakban gyakran elhagyjuk a nevezőt, és K helyett

a $\mathbb{Z}[\theta]$ gyűrűt vizsgáljuk, amelynek $\beta \in \mathbb{Z}[\theta]$ elemei a következő alakban írhatók fel:

$$\beta = b_0 + b_1\theta + b_2\theta^2 + \ldots + b_{m-1}\theta^{m-1} \qquad (b_i \in \mathbb{Z}). \tag{2.1.3}$$

A K testtől való függést legtöbbször két paraméterrel fogjuk kifejezni: az m fokszámmal és az alább definiált F mennyiséggel, amely az f együtthatóinak méretét méri. Legyen $H(f)$ (f magassága) és F a következő:

$$H(f) := \max_{i=0}^{m} |f_i|,$$

$$F := \log(H(f) + 1).$$

Megjegyzés: ebben a dolgozatban a logaritmusok alapszámát legtöbbször nem tüntetjük fel, de mindvégig ugyanarról a rögzített alapról van szó. Gondolhatunk kettes alapú logaritmusra, mint az ábrázoláshoz szükséges bitek számára, de minden számításunk érvényes tetszőleges rögzített ≥ 2 alapra, hiszen az alapszám változtatása csak konstanssal való szorzást jelent.

A számításokhoz szükségünk lesz a θ, m és F közötti következő két egyenlőtlenségre:

2.1.12. Lemma.

$$\log|\theta| < F, \tag{2.1.4}$$

$$\log\bigl(1 + |\theta| + |\theta|^2 + \ldots + |\theta|^{m-1}\bigr) < mF. \tag{2.1.5}$$

Bizonyítás. A Cauchy-egyenlőtlenség alapján $|\theta| < H(f) + 1$ (lásd pl. [18, 323. o.]), ami ekvivalens az első állítással. A második ebből következik:

$$1 + |\theta| + |\theta|^2 + \ldots + |\theta|^{m-1} \leq 1 + (H(f)+1) + (H(f)+1)^2 + \ldots + (H(f)+1)^{m-1} =$$
$$= \frac{(H(f)+1)^m - 1}{(H(f)+1) - 1} < (H(f)+1)^m.$$

■

2.2. Alapvető műveletek

Ebben a szakaszban az algebrai számtestben végzett alapvető műveletek hatékonyságát vizsgáljuk. A négy alapműveleten kívül megnézünk néhány kifejezetten $\mathbb{R}$-beli műveletet is, mint a kisebb-nagyobb összehasonlítás és az egészre kerekítés. A szakaszt két részre bontjuk: először (2.2.1. szakasz) megvizsgáljuk, hogy az egyes műveletek során mennyivel nőhet a számok mérete (az ábrázoláshoz szükséges memóriát nézve), azután pedig (2.2.2. szakasz) a műveletek futási idejét számítjuk ki.

A témában vannak korábbi eredmények: [8, 5. szakasz] és [9, 3. szakasz], de [8] csak a szorzással foglalkozik, és azt sem a számunkra megfelelő formában (pl. ugyanazt a korlátot használja a minimálpolinomra és az ábrázolási együtthatókra, pedig az előbbi konstans, az utóbbi pedig változó), [9] pedig nem kellően általános (pl. kizárólag gyorsszorzást vizsgál, ami csak nagy egész számok esetén hatékony). Egy-két eredményt és ötletet használni tudunk ezekből a munkákból (és ahol ezt tesszük, azt mindig feltüntetjük), de egyébként – legjobb tudomásunk szerint – az itt leírtakhoz hasonló általános eredményeket mások még nem számoltak ki, a kifejezetten $\mathbb{R}$-beli műveleteket pedig algebrai számokra még egyáltalán nem vizsgálták.

2.2.1. Méretnövekedés a műveletek során

Egy $\alpha \in \mathbb{Z}[\theta]$ algebrai számnak, melyet $\alpha = a_0 + a_1\theta + a_2\theta^2 + \ldots + a_{m-1}\theta^{m-1}$ $(a_i \in \mathbb{Z})$ alakban írunk fel, a következő függvénnyel mérjük a méretét:

$$\mathrm{s}(\alpha) := \begin{cases} \log \max_{i=0}^{m-1} |a_i| & (\alpha \neq 0), \\ 0 & (\alpha = 0). \end{cases} \tag{2.2.1}$$

Ez a mennyiség – a rögzített m fokszámmal együtt – leírja az α tárolásához szükséges tárterület méretét. Az $\mathrm{s}(\cdot)$ operátor egész számokra leegyszerűsödik: ha n egész, akkor $\mathrm{s}(n) = \log|n|$, ha $n \neq 0$, különben 0. Definiáljuk továbbá a többváltozós $\mathrm{s}(\cdot)$-et is: $\mathrm{s}(\alpha_1, \alpha_2, \ldots, \alpha_n) := \max_{i=1}^{n} \mathrm{s}(\alpha_i)$.

Használni fogjuk [9] néhány eredményét. A szerzők az $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$ bázis szerinti kifejtést használják, ahol $\omega_1 = 1$. Az eredményeikben három Ω-tól függő konstans, C_1, C_2 és C_3 szerepel. Az első kettőt a következő állítás segítségével vezetik be [9, (1)], ez a mi jelöléseinkkel így néz ki:

2.2.1. Lemma. *Legyen $\alpha \in \mathbb{Z}[\theta]$, legyen az α együtthatóvektora $x := (a_0, a_1, \ldots, a_{m-1})^T$, jelölje f gyökeit $\theta_1, \theta_2, \ldots, \theta_m$, legyen $\alpha_i := a_0 + a_1\theta_i + \ldots + a_{m-1}\theta_i^{m-1}$, valamint legyen $y := (\alpha_1, \alpha_2, \ldots, \alpha_m)^T$. Ekkor létezik olyan $C_1, C_2 > 0$, hogy*

$$\frac{1}{C_2}\|x\|_\infty \leq \|y\|_2 \leq C_1\|x\|_\infty.$$

A harmadik konstanst az $(\omega_i\omega_j)_{i,j}$ szorzási táblázat együtthatóiból definiálják:

$$\omega_i\omega_j = m_{i,j,1}\omega_1 + m_{i,j,2}\omega_2 + \ldots + m_{i,j,m}\omega_m,$$

$$C_3 := \max_{i,j,k} |m_{i,j,k}|.$$

Ahhoz, hogy az eredményeiket használni tudjuk, ki kell fejezni a három konstansukat

az általunk bevezetett m és F paraméterekkel. Erről szól a következő három becslés:

2.2.2. Lemma.

$$\log C_1 < mF + \frac{1}{2}\log m, \tag{2.2.2}$$

$$\log C_2 < \binom{m}{2}F + m\log m, \tag{2.2.3}$$

$$\log C_3 < (m-1)F. \tag{2.2.4}$$

Bizonyítás. Az x és az y kapcsolatát (lásd a 2.2.1. Lemmát) az $y = Vx$ szorzással fejezhetjük ki, ahol V a következő Vandermonde-mátrix:

$$V = \begin{pmatrix} 1 & \theta_1 & \theta_1^2 & \dots & \theta_1^{m-1} \\ 1 & \theta_2 & \theta_2^2 & \dots & \theta_2^{m-1} \\ \vdots & \vdots & & & \vdots \\ 1 & \theta_m & \theta_m^2 & \dots & \theta_m^{m-1} \end{pmatrix}.$$

Mivel $\|y\|_\infty \le \|y\|_2 \le \sqrt{m}\,\|y\|_\infty$, ezért C_1-et és C_2-t választhatjuk a következőképpen:

$$C_1 := \sqrt{m}\,\|V\|_\infty, \qquad C_2 := \|V^{-1}\|_\infty,$$

így már csak $\|V\|_\infty$-re és $\|V^{-1}\|_\infty$-re kell becslést adni.

Az előbbihez (2.1.5)-öt használjuk, és ezzel megmutatjuk (2.2.2)-t:

$$\log\|V\|_\infty = \log\max_{i=1}^{m}(1 + |\theta_i| + |\theta_i|^2 + \ldots + |\theta_i|^{m-1}) < mF.$$

A (2.2.3)-hoz szükségünk van $\|V^{-1}\|_\infty$-ra. V inverze kifejezhető $(V^{-1})_{ij} = C_{ji}/\det V$ alakban, ahol C_{ij} a V-ből az i. sor és j. oszlop elhagyásával kapott aldetermináns, megszorozva $(-1)^{i+j}$-nel. Ismert, hogy a Vandermonde-mátrix determinánsa a következőképpen írható fel (lásd pl. [18, 185. o.]):

$$\det V = \prod_{i=1}^{m}\prod_{j=i+1}^{m}(\theta_j - \theta_i),$$

és az is, hogy ennek négyzete az f diszkriminánsa (lásd pl. [14, Prop. 3.3.5.]). Az utóbbi az f együtthatóinak polinomiális függvénye, tehát egész szám, és mivel a Vandermonde-determináns nem nulla, ezért $|\det V| \ge 1$, következésképp $|(V^{-1})_{ij}| \le |C_{ji}|$.

C_{ji} egy $(m-1)\times(m-1)$-es determináns, amelynek a kifejtésében (lásd (2.2.16)-ot) $(m-1)!$ tag van, és mindegyik tag $m-1$ darab V-beli elem szorzata. Ezek az elemek $V_{kl} = \theta_k^{l-1}$ alakúak, és mivel minden szorzat különböző oszlopokból való elemekből áll, ezért a kitevők összege legfeljebb $0 + 1 + 2 + \ldots + (m-1) = \binom{m}{2}$ lehet. A (2.1.4) miatt $\log|V_{kl}| < (l-1)F$, következésképp $\log|C_{ji}| < \binom{m}{2}F + \log(m-1)!$. Ebből már megkapjuk (2.2.3)-at:

$$\log C_2 = \log \|V^{-1}\|_\infty = \log \max_{i=1}^{m} \big(\big|(V^{-1})_{i1}\big| + \big|(V^{-1})_{i2}\big| + \ldots + \big|(V^{-1})_{im}\big| \big) \leq$$
$$\leq \log \max_{i=1}^{m} (|C_{1i}| + |C_{2i}| + \ldots + |C_{mi}|) <$$
$$< \binom{m}{2} F + \log(m-1)! + \log m \leq \binom{m}{2} F + m \log m.$$

A (2.2.4)-hez θ^{m+k}-t minden $k = 0, 1, \ldots, m-2$-re kifejezzük θ alacsonyabb hatványainak segítségével:

$$\theta^{m+k} = r_{k,0} + r_{k,1}\theta + r_{k,2}\theta^2 + \ldots + r_{k,m-1}\theta^{m-1}, \tag{2.2.5}$$

és ekkor $C_3 = \max_{k=0}^{m-2} \max_{j=0}^{m-1} |r_{k,j}|$.

Az $f(\theta) = 0$ egyenlőség alapján fel tudunk írni egy rekurzív formulát $r_{k,l}$-re (lásd pl. [14, 159. o.]):

$$r_{0,l} = -f_l, \qquad r_{k+1,l} = r_{k,l-1} - f_l r_{k,m-1}, \tag{2.2.6}$$

ahol $r_{k,-1} = 0$. Ebből indukcióval könnyű megmutatni, hogy

$$|r_{k,l}| < (H(f) + 1)^{k+1}, \tag{2.2.7}$$

és innen már következik (2.2.4). ■

A következőkben azzal foglalkozunk, hogy mennyire növekedhet $\mathrm{s}(\cdot)$ az egyes alapvető műveletek során. Kezdjük a triviális állításokkal:

2.2.3. Lemma.

$$\alpha, \beta \in \mathbb{Z}[\theta]: \qquad \mathrm{s}(\alpha \pm \beta) \leq \mathrm{s}(\alpha, \beta) + \log 2, \tag{2.2.8}$$

$$n \in \mathbb{Z}^+,\ \alpha_1, \ldots, \alpha_n \in \mathbb{Z}[\theta]: \qquad \mathrm{s}(\alpha_1 + \ldots + \alpha_n) \leq \mathrm{s}(\alpha_1, \ldots, \alpha_n) + \log n, \tag{2.2.9}$$

$$n \in \mathbb{Z},\ \alpha \in \mathbb{Z}[\theta]: \qquad \mathrm{s}(n\alpha) \leq \mathrm{s}(\alpha) + \mathrm{s}(n). \tag{2.2.10}$$

Folytassuk a multiplikatív műveletekkel, a szorzással és az osztással. [9, Prop. 4 előtt] bebizonyítja a következőt. (Megjegyezzük, hogy [9] kissé eltérő jelölést használ a számok méretére: az ottani $\mathrm{S}(\alpha) := m\,\mathrm{s}(\alpha)$, de itt átírtuk azokat $\mathrm{s}(\cdot)$-re.)

2.2.4. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta]$, akkor*

$$\mathrm{s}(\alpha\beta) \leq \mathrm{s}(\alpha) + \mathrm{s}(\beta) + \log C_3 + 2 \log m,$$

és ha $\alpha \in \mathbb{Z}[\theta]$, $\alpha \neq 0$ és $\alpha^{-1} = \beta/n$, ahol $\beta \in \mathbb{Z}[\theta]$ és $n \in \mathbb{Z}$, akkor

$$\mathrm{s}(\beta) \leq (m-1)\,\mathrm{s}(\alpha) + (m-1)\log C_1 + \log C_2 + \frac{1}{2}\log m,$$
$$\mathrm{s}(n) \leq m\,\mathrm{s}(\alpha) + m \log C_1 - \frac{m}{2}\log m.$$

Ezekből, ha behelyettesítjük a 2.2.2. Lemmában C_1-re, C_2-re és C_3-ra kiszámított

becsléseinket, a következőket kapjuk:

2.2.5. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta]$, akkor*

$$\mathrm{s}(\alpha\beta) \leq \mathrm{s}(\alpha) + \mathrm{s}(\beta) + (m-1)F + 2\log m, \tag{2.2.11}$$

és ha $\alpha \in \mathbb{Z}[\theta]$, $\alpha \neq 0$ és $\alpha^{-1} = \beta/n$, ahol $\beta \in \mathbb{Z}[\theta]$ és $n \in \mathbb{Z}$, akkor

$$\mathrm{s}(\beta) \leq (m-1)\,\mathrm{s}(\alpha) + \frac{3}{2}m(m-1)F + \frac{3}{2}m\log m, \tag{2.2.12}$$

$$\mathrm{s}(n) \leq m\,\mathrm{s}(\alpha) + m^2 F. \tag{2.2.13}$$

A következő eredmény egy alsó és egy felső korlát az algebrai egészekre az együtthatóméret függvényében.

2.2.6. Lemma. *Ha $\alpha \in \mathbb{Z}[\theta]$ és $\alpha \neq 0$, akkor*

$$\log|\alpha| < \mathrm{s}(\alpha) + mF, \tag{2.2.14}$$

$$\log|\alpha^{-1}| < (m-1)\,\mathrm{s}(\alpha) + \frac{3}{2}m^2(F+1). \tag{2.2.15}$$

Bizonyítás. (2.2.14) következik (2.1.5)-ből:

$$\log|\alpha| = \log\left|\sum_{j=0}^{m-1} a_j\theta^j\right| \leq \mathrm{s}(\alpha) + \log\sum_{j=0}^{m-1}|\theta|^j < \mathrm{s}(\alpha) + mF.$$

(2.2.15) kijön (2.2.14)-ből és (2.2.12)-ből:

$$\log|\alpha^{-1}| = \log\frac{|\beta|}{|n|} \leq \log|\beta| < \mathrm{s}(\beta) + mF \leq$$
$$\leq (m-1)\,\mathrm{s}(\alpha) + m\frac{3m-1}{2}F + \frac{3}{2}m\log m,$$

és egyszerűsítések után megkapjuk (2.2.15)-öt. ■

Néha szükségünk lesz algebrai egészekből álló determinánsokra, ezért levezetünk rá egy formulát a fentiek segítségével.

2.2.7. Lemma. *Legyen $A \in \mathbb{Z}[\theta]^{n\times n}$ egy mátrix az α_{ij} elemekkel. Ekkor*

$$\mathrm{s}(\det A) \leq n\max_{i,j}\mathrm{s}(\alpha_{ij}) + (n-1)((m-1)F + 2\log m) + n\log n.$$

Bizonyítás. A determináns teljes kifejtése a következő:

$$\det A = \sum_{\sigma\in S_n}(-1)^{N(\sigma)}\prod_{i=1}^{n}\alpha_{i,\sigma(i)}, \tag{2.2.16}$$

ahol S_n az $\{1, 2, \ldots, n\}$ halmaz összes $(n!)$ permutációja, és $N(\sigma)$ a σ inverziószáma. Felhasználva (2.2.9)-et:

$$\mathrm{s}(\det A) \leq \max_{\sigma\in S_n}\mathrm{s}\left(\prod_{i=1}^{n}\alpha_{i,\sigma(i)}\right) + \log n!,$$

és (2.2.11) ismételt alkalmazásával, valamint az $n! \leq n^n$ becsléssel adódik az állítás. ■

2.2.2. A műveletek futási ideje

Ebben a szakaszban az algebrai számtestben végzett műveletek futási idejét vizsgáljuk. A nagyságrendi becslések megfogalmazásához az $O(\cdot)$ jelölést használjuk:

2.2.8. Definíció. Az $f : \mathbb{R}^+ \to \mathbb{R}$ és $g : \mathbb{R}^+ \to \mathbb{R}_0^+$ függvényekre $f(x) = O(g(x))$, ha

$$\exists M, c > 0 : \forall x \geq c : |f(x)| \leq Mg(x).$$

A futási időt sok esetben több paraméter segítségével fejezzük ki, ekkor az $O(\cdot)$ többváltozós általánosítását használjuk:

2.2.9. Definíció. Az $f : (\mathbb{R}^+)^n \to \mathbb{R}$ és $g : (\mathbb{R}^+)^n \to \mathbb{R}_0^+$ többváltozós függvényekre $f(x) = O(g(x))$, ha

$$\exists M, c > 0 : \forall x \in (\mathbb{R}^+)^n : \|x\|_\infty \geq c \implies |f(x)| \leq Mg(x).$$

Mivel az algebrai számokat egész számokkal ábrázoljuk, ezért szükségünk lesz az egészek közötti műveletek bonyolultságára. Ezek közül a szorzás és az osztás a különösen érdekes, mert azokra több különböző algoritmus létezik különböző futási időkkel. Az általánosság kedvéért a következő jelölést fogjuk használni.

Legyen $\mathrm{Mul}(A, B)$ az a és b egész számok összeszorzásának futási ideje, ahol A és B a két szám hosszának korlátja (azaz $\mathrm{s}(a) \leq A$ és $\mathrm{s}(b) \leq B$), és legyen $\mathrm{Mul}(A) := \mathrm{Mul}(A, A)$. A kifejezés értéke a szorzási algoritmustól függ, például:

- elemi szorzás: $\mathrm{Mul}(A, B) = O(AB)$,
- Karatsuba-szorzás: $\mathrm{Mul}(A) = O(A^{\log_2 3})$,
- Schönhage–Strassen-szorzás: $\mathrm{Mul}(A) = O(A \log A \log\log A)$.

Hangsúlyozzuk, hogy ezek aszimptotikus értékek, és pl. a Schönhage–Strassen-szorzás csak nagy számok esetén lesz gyorsabb, mint a normál szorzás. A szorzási algoritmusokról részletesen pl. [20, 4.3.3.]-ban van szó.

A $\mathrm{Mul}(\cdot)$ függvényről itt csak annyit teszünk fel, hogy teljesíti a következő feltételeket minden $A, B, C > 0$ számra:

$$\mathrm{Mul}(A,\, B) = \mathrm{Mul}(B,\, A),$$

$$A \leq B \implies \mathrm{Mul}(A,\, C) \leq \mathrm{Mul}(B,\, C),$$

$$\mathrm{Mul}(A + B,\, C) \leq \mathrm{Mul}(A,\, C) + \mathrm{Mul}(B,\, C),$$

$$\mathrm{Mul}(AB,\, C) \leq A\,\mathrm{Mul}(B,\, C),$$

$$\mathrm{Mul}(AB) \geq A\,\mathrm{Mul}(B).$$

Feltesszük továbbá, hogy az a/b egész maradékos osztás $O(\mathrm{Mul}(\mathrm{s}(b), \mathrm{s}(q)))$ idő alatt számítható ki, ahol q az osztás hányadosa (lásd pl. [20, 4.3.3. D]).

Jelöljük $\mathrm{T}(F)$-fel az F kifejezés kiszámításához szükséges gépi aritmetikai műveletek számát (rögzített gépen, a gép beépített fix méretű számaira).

Kezdjük a triviális műveletekkel:

2.2.10. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta]$ és $n \in \mathbb{Z}$, akkor*

$$\mathrm{T}(\alpha \pm \beta) = O(m\,\mathrm{s}(\alpha, \beta))), \tag{2.2.17}$$

$$\mathrm{T}(n\alpha) = O(m\,\mathrm{Mul}(\mathrm{s}(n), \mathrm{s}(\alpha))). \tag{2.2.18}$$

Algebrai számok szorzására a következő eredményt tudjuk adni (ez az állítás is, és az összes többi, ahol nem jelezzük másként, a szerző saját számítása):

2.2.11. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta]$, akkor*

$$\mathrm{T}(\alpha\beta) = O\big(m^2\,\mathrm{Mul}(\mathrm{s}(\alpha), \mathrm{s}(\beta)) \;+\; m^2\,\mathrm{Mul}(\mathrm{s}(\alpha) + \mathrm{s}(\beta) + \log m,\; mF)\big). \tag{2.2.19}$$

Bizonyítás. Legyen $\gamma := \alpha\beta$, és jelöljük α, β és γ együtthatóit rendre a_i-vel, b_i-vel és c_i-vel. A szorzást a következőképpen végezzük el:

1. Kiszámítjuk a szorzatot úgy, mintha θ egy változó lenne, azaz polinomként szorozzuk össze őket. A szorzat együtthatói a következők lesznek:
$$d_l := \sum_j a_j b_{l-j} \qquad (0 \le l \le 2m-2).$$
2. Kiszámítjuk a polinom maradékát modulo f. Az eredmény együtthatói:
$$c_l := d_l + \sum_{k=0}^{m-2} d_{m+k} r_{k,l} \qquad (0 \le l \le m-1),$$
ahol az $r_{k,l}$-eket (2.2.5)-ben definiáltuk, és előre ki tudjuk számítani f-ből a (2.2.6) rekurzív képlettel.

Mindkét lépés fő művelete a kb. m^2 szorzás, ezért a teljes futási idő:

$$\mathrm{T}(\alpha\beta) = O\Bigg(m^2\,\mathrm{Mul}(\mathrm{s}(\alpha), \mathrm{s}(\beta)) + m^2\,\mathrm{Mul}(\max_l \mathrm{s}(d_l), \max_{k,l} \mathrm{s}(r_{k,l}))\Bigg).$$

A második $\mathrm{Mul}(\cdot)$-ban a változók méretének kiszámítására a d_l fenti definícióját, valamint az $r_{k,l}$-re korábban kiszámított (2.2.7) becslést használjuk:

$$\mathrm{s}(d_l) \le \mathrm{s}(\alpha) + \mathrm{s}(\beta) + \log m,$$

$$\mathrm{s}(r_{k,l}) < (k+1)F < mF,$$

és ezeket behelyettesítve megkapjuk a bizonyítandó állítást. ■

A $\gamma = \alpha\beta$ szorzásra [9] egy másik módszert ad: számítsuk ki α-ból azt az M_α mátrixot, amely β-t γ-ba transzformálja, azaz amelyre $M_\alpha(b_0, b_1, \ldots, b_{m-1})^T = (c_0, c_1, \ldots, c_{m-1})^T$. Esetünkben a fenti formulákból könnyen levezethető, hogy

$$(M_\alpha)_{i,j} = a_{i-j} + \sum_{k=0}^{j-2} a_{m+k-j+1} r_{k,i-1} \tag{2.2.20}$$

(ahol a_{i-j} alatt 0-t értünk, ha $i < j$). Az M_α kiszámításához tehát

$$\mathrm{T}(M_\alpha) = O\big(m^3 \operatorname{Mul}(\mathrm{s}(\alpha), mF)\big) \tag{2.2.21}$$

idő szükséges. Ez lassabb, mint a fenti módszer szerinti szorzás (2.2.19), de ha gyakran szorzunk ugyanazzal az α-val, akkor M_α-t előre kiszámíthatjuk, és akkor már csak egy mátrix-vektor szorzást kell elvégezni, amely valamelyest gyorsabb:

$$\mathrm{T}(\alpha\beta)_{M_\alpha} = O\big(m^2 \operatorname{Mul}(\mathrm{s}(\alpha) + mF, \mathrm{s}(\beta))\big).$$

Az M_α mátrixot használhatjuk még egy dologra: az α inverzének kiszámításához.

2.2.12. Lemma. *Ha $\alpha \in \mathbb{Z}[\theta]$ és $\alpha \neq 0$, akkor*

$$\mathrm{T}(\alpha^{-1}) = O\big(m^3 \operatorname{Mul}(m\,\mathrm{s}(\alpha) + m^2 F)\big). \tag{2.2.22}$$

Bizonyítás. A bizonyítás ötlete [9, Prop. 10]-ből származik, de itt az általánosabb $\operatorname{Mul}(\cdot)$ függvényt és a precízebb $O(\cdot)$ jelölést használjuk. Az $\alpha^{-1} = \beta/n$ kiszámítása a következőképpen történik:

1. Kiszámítjuk az M_α mátrixot (2.2.20) alapján.
2. Megoldjuk az $M_\alpha(b_0/n, b_1/n, \ldots, b_{m-1}/n)^T = (1, 0, \ldots, 0)$ egyenletrendszert β/n-re.

Az utóbbi elvégezhető a Bareiss-algoritmussal [7] (amelyről részletesebben a 2.3. szakaszban lesz szó) a következő idő alatt:

$$\mathrm{T}(\text{Bareiss}) = O\big(m^3 \operatorname{Mul}(m \log(mA))\big),$$

ahol A a mátrix elemeinek korlátja. Ezt az M_α (2.2.20) formulájából kiszámíthatjuk:

$$\log A = \log \max_{i,j} |(M_\alpha)_{i,j}| = O(\mathrm{s}(\alpha) + mF),$$

amelyet ha behelyettesítünk az előző képletbe, akkor megkapjuk (2.2.22) jobb oldalát, és a hátralévő $\mathrm{T}(M_\alpha)$ (2.2.21) hozzáadása nem változtat a nagyságrenden. ■

Következményként megkapjuk az osztás futási idejét is:

2.2.13. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta]$ és $\beta \neq 0$, akkor*

$$\mathrm{T}\left(\frac{\alpha}{\beta}\right) = O\big(m^3 \operatorname{Mul}(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2 F)\big). \tag{2.2.23}$$

Bizonyítás. Ha $\beta^{-1} = \frac{\gamma}{n}$ ($n \in \mathbb{Z}$), akkor $\frac{\alpha}{\beta} = \frac{\alpha\gamma}{n}$. (2.2.12)-ből $\mathrm{s}(\gamma) = O(m\,\mathrm{s}(\beta) + m^2 F)$, és $\mathrm{T}(\beta^{-1})$ (2.2.22) és $\mathrm{T}(\alpha\gamma)$ (2.2.19) összeadásával megkapjuk az állítást. ■

A szakasz további részében néhány kizárólag valós számokra vonatkozó művelettel fogunk foglalkozni. Feltesszük tehát, hogy $\theta \in \mathbb{R}$, és így $K \subset \mathbb{R}$.

A számításokhoz szükségünk lesz a következő segédállításra, amely megmondja, hogy a $\mathbb{Z}[\theta]$-beli elemek közelítéséhez mennyire pontosan kell θ-t approximálni.

2.2.14. Lemma. *Legyen $\alpha, \beta \in \mathbb{Z}[\theta] \setminus \{0\}$. Legyen $\tilde{\theta} \in \mathbb{R}$ olyan közel θ-hoz, hogy*

$$\log |\tilde{\theta} - \theta|^{-1} \geq \mathrm{s}(\alpha) + (m-1)\,\mathrm{s}(\beta) + \frac{3}{2}m(m+1)(F+1), \tag{2.2.24}$$

és legyen $\tilde{\alpha} = a_0 + a_1\tilde{\theta} + a_2\tilde{\theta}^2 + \ldots + a_{m-1}\tilde{\theta}^{m-1}$. Ekkor

$$|\tilde{\alpha} - \alpha| < |\beta|.$$

Bizonyítás. Először (2.2.24)-ből csak annyit használunk, hogy $m|\tilde{\theta} - \theta| < \frac{1}{2}$.

$$\begin{aligned}
|\tilde{\alpha} - \alpha| &= \left| \sum_{k=0}^{m-1} a_k \left(\Big(\theta + (\tilde{\theta} - \theta)\Big)^k - \theta^k \right) \right| = \left| \sum_{k=0}^{m-1} a_k \sum_{j=1}^{k} \binom{k}{j} \theta^{k-j} (\tilde{\theta} - \theta)^j \right| \leq \\
&\leq \sum_{k=0}^{m-1} |a_k| \sum_{j=1}^{k} \binom{k}{j} |\theta|^{k-j} |\tilde{\theta} - \theta|^j = \sum_{j=1}^{m-1} \sum_{l=0}^{m-j-1} |a_{l+j}| \binom{l+j}{j} |\theta|^l |\tilde{\theta} - \theta|^j \leq \\
&\leq \Big(\max_{i=0}^{m-1} |a_i|\Big) \sum_{j=1}^{m-1} m^j |\tilde{\theta} - \theta|^j \sum_{l=0}^{m-j-1} |\theta|^l < \Big(\max_{i=0}^{m-1} |a_i|\Big) 2m|\tilde{\theta} - \theta| \sum_{l=0}^{m-1} |\theta|^l.
\end{aligned}$$

Logaritmust véve és alkalmazva (2.1.5)-öt:

$$\log |\tilde{\alpha} - \alpha| < \mathrm{s}(\alpha) + \log 2m|\tilde{\theta} - \theta| + mF \leq \mathrm{s}(\alpha) + \frac{3}{2}m(F+1) + \log |\tilde{\theta} - \theta|.$$

Adjuk hozzá ehhez az egyenlőtlenséghez (2.2.15)-öt β-val:

$$\log |\tilde{\alpha} - \alpha| + \log |\beta^{-1}| < \mathrm{s}(\alpha) + (m-1)\,\mathrm{s}(\beta) + \frac{3}{2}m(m+1)(F+1) + \log |\tilde{\theta} - \theta|.$$

A (2.2.24) alapján a jobb oldal ≤ 0, tehát a bal oldal < 0, ami ekvivalens az állítással. ■

Most következzenek a műveletek, és a futási idejük.

2.2.15. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta] \subset \mathbb{R}$, akkor*

$$\mathrm{T}(\alpha < \beta) = \mathrm{T}(\alpha \leq \beta) = O\big(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha, \beta) + m^2 F)\big). \tag{2.2.25}$$

Bizonyítás. Mivel $\alpha < \beta$ ekvivalens azzal, hogy $\alpha - \beta < 0$, ezért elég csak az $\alpha < 0$ összehasonlítással foglalkozni, és hasonló igaz $\leq$-re is. Tegyük fel, hogy $\alpha \neq 0$, ellenkező esetben a művelet triviális.

A θ-t közelítjük egy $\tilde{\theta} = \frac{u}{d}$ racionális számmal, ahol $d \in \mathbb{Z}^+$ és u a $\lfloor \theta d \rfloor$ és a $\lceil \theta d \rceil$ közül a kisebb abszolút értékű. Olyan nagy d-re van szükségünk, hogy $\tilde{\theta}$ olyan közel legyen

θ-hoz, hogy az $\tilde{\alpha} = a_0 + a_1\tilde{\theta} + a_2\tilde{\theta}^2 + \ldots + a_{m-1}\tilde{\theta}^{m-1}$ és a pontos α azonos előjelű legyen. Ezt garantálni tudjuk, ha $|\tilde{\alpha} - \alpha| < |\alpha|$, mert akkor

$$\alpha > 0 \implies \tilde{\alpha} \geq \alpha - |\tilde{\alpha} - \alpha| > 0,$$

$$\alpha < 0 \implies \tilde{\alpha} \leq \alpha + |\tilde{\alpha} - \alpha| < 0.$$

Ehhez a 2.2.14. Lemma alapján ($\beta := \alpha$-val) egy olyan d-t kell választani, amelyre

$$\log d \geq m\,\mathrm{s}(\alpha) + \frac{3}{2}m(m+1)(F+1),$$

mert $\log|\tilde{\theta} - \theta|^{-1} > \log d$. Választhatjuk d-t a legkisebb ilyen egész számnak, de hatékonyabb, ha felkerekítjük a legközelebbi kettőhatványra. Mindkét esetben:

$$\log d = O\big(m\,\mathrm{s}(\alpha) + m^2F\big). \tag{2.2.26}$$

Az $\tilde{\alpha}$-ot felírhatjuk a következőképpen:

$$\tilde{\alpha} = a_0 + a_1\frac{u}{d} + \ldots + a_{m-1}\left(\frac{u}{d}\right)^{m-1} = \frac{a_0\,d^{m-1} + a_1u\,d^{m-2} + \ldots + a_{m-1}u^{m-1}}{d^{m-1}}.$$

Az előjel csak a számlálótól függ, amelyet jelöljünk r-rel, és kiszámíthatjuk a következő rekurzióval:

$$r_0 := 0, \qquad r_{k+1} := r_ku + a_{m-k-1}d^k, \qquad r := r_m.$$

Ha d-t kettőhatványnak választottuk, akkor az egyetlen érdemi művelet az r_ku szorzás.

Az u és az r_k méretét a következőképpen tudjuk becsülni (az utóbbit indukcióval):

$$|u| = \min(|\lfloor\theta d\rfloor|, |\lceil\theta d\rceil|) \leq |\theta d| = |\theta|d,$$

$$|r_k| \leq \Big(\max_{i=0}^{m-1}|a_i|\Big)d^{k-1}\big(1 + |\theta| + \ldots + |\theta|^{k-1}\big).$$

A (2.1.5) és a (2.2.26) felhasználásával:

$$\begin{aligned}\mathrm{T}(r_ku) &= \mathrm{Mul}(\mathrm{s}(r_k), \mathrm{s}(u)) \leq \mathrm{Mul}(\mathrm{s}(\alpha) + m\log d + mF,\ F + \log d) = \\ &= O(\mathrm{Mul}(m^2\,\mathrm{s}(\alpha) + m^3F,\ m\,\mathrm{s}(\alpha) + m^2F)) = O(m\,\mathrm{Mul}(m\,\mathrm{s}(\alpha) + m^2F)),\end{aligned}$$

és a teljes számítás m-szer végez ilyen szorzást. ■

2.2.16. Lemma. *Ha $\alpha, \beta \in \mathbb{Z}[\theta] \subset \mathbb{R}$, $n \in \mathbb{Z}$ és $\beta \neq 0$, $n \neq 0$, akkor*

$$\mathrm{T}\left(\left\lfloor\frac{\alpha}{n}\right\rfloor\right) = \mathrm{T}\left(\left\lceil\frac{\alpha}{n}\right\rceil\right) = \mathrm{T}\left(\left\lfloor\frac{\alpha}{n}\right\rfloor\right) = O\big(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha, n) + m^2F)\big), \tag{2.2.27}$$

$$\mathrm{T}\left(\left\lfloor\frac{\alpha}{\beta}\right\rfloor\right) = \mathrm{T}\left(\left\lceil\frac{\alpha}{\beta}\right\rceil\right) = \mathrm{T}\left(\left\lfloor\frac{\alpha}{\beta}\right\rfloor\right) = O\big(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha) + m^2\,\mathrm{s}(\beta) + m^3F)\big). \tag{2.2.28}$$

Bizonyítás. Mivel $\lfloor\alpha/n\rceil = \lfloor\alpha/n + 1/2\rfloor$, ezért csak az első két kerekítéssel ($\lfloor\cdot\rfloor$ és $\lceil\cdot\rceil$) kell foglalkozni. Ha $\alpha \in \mathbb{Z}$, akkor a művelet egy egyszerű maradékos osztás, amelyet $\mathrm{Mul}(\mathrm{s}(\alpha), \mathrm{s}(n))$ idő alatt elvégezhetünk, ezért a továbbiakban tegyük fel, hogy $\alpha \notin \mathbb{Z}$.

A (2.2.27) bizonyítása hasonló, mint (2.2.25)-é. Itt a d nevezőnek olyan nagynak kell lennie, hogy $\tilde{\alpha}/n$-nek és α/n-nek ugyanaz legyen az egészrésze. Ezt a következő feltétellel tudjuk garantálni:

$$\left|\frac{\tilde{\alpha}}{n} - \frac{\alpha}{n}\right| < \min\Big(\Big|\frac{\alpha}{n} - \Big\lfloor\frac{\alpha}{n}\Big\rfloor\Big|, \Big|\frac{\alpha}{n} - \Big\lceil\frac{\alpha}{n}\Big\rceil\Big|\Big).$$

Ezt szorozzuk fel $|n|$-nel, és használjuk a 2.2.14. Lemmát $\beta \in \big\{\alpha - n\lfloor\alpha/n\rfloor,\ \alpha - n\lceil\alpha/n\rceil\big\}$ helyettesítéssel. Ehhez először adni kell egy becslést $\mathrm{s}(\beta)$-ra. Mivel $\beta - \alpha$ egész szám, ezért α és β együtthatói a konstans tag kivételével megegyeznek, így $\mathrm{s}(\beta) \le \max(\mathrm{s}(\alpha), \mathrm{s}(b_0))$. β konstans tagját a következőképpen tudjuk becsülni (mivel $|\beta/n| < 1$, ezért $|\beta| < |n|$):

$$|b_0| = |a_0 - \alpha + \beta| < |\alpha - a_0| + |n| = \left|\sum_{k=1}^{m-1} a_k\theta^k\right| + |n| \le \max\Big(|n|, \max_{i=1}^{m-1}|a_i|\Big)\left(\sum_{i=0}^{m-1}|\theta|^i\right),$$

amelynek a logaritmusa $\le \mathrm{s}(\alpha, n) + mF$, ezért

$$\mathrm{s}(\beta) \le \mathrm{s}(\alpha, n) + mF.$$

Alkalmazva a 2.2.14. Lemmát, d-t választhatjuk úgy, hogy

$$\log d = O\big(m\,\mathrm{s}(\alpha, n) + m^2F\big).$$

Ezután az előző lemmához hasonlóan kiszámítjuk r-re, az $\tilde{\alpha}$ számlálójára, hogy

$$\mathrm{T}(r) = O\big(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha, n) + m^2F)\big),$$
$$\mathrm{s}(r) = O\big(m^2\,\mathrm{s}(\alpha, n) + m^3F\big).$$

Utolsó lépésként osztani kell n-nel, amelynek, mint maradékos osztásnak, a futási ideje legfeljebb $\mathrm{Mul}(m^2\,\mathrm{s}(\alpha, n) + m^3F,\ \mathrm{s}(n))$, és ezt hozzáadva $\mathrm{T}(r)$-hez megkapjuk (2.2.27)-et.

A (2.2.28)-hoz először kiszámítjuk α/β-t $\alpha\gamma/n$ ($n \in \mathbb{Z}$) alakban. Ez (2.2.23) alapján

$$\mathrm{T}(\alpha/\beta) = O\big(m^3\,\mathrm{Mul}(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2F)\big)$$

idő alatt megtehető. A paraméterek mérete (2.2.12), (2.2.11) és (2.2.13) alapján:

$$\mathrm{s}(\gamma) = O\big(m\,\mathrm{s}(\beta) + m^2F\big),$$
$$\mathrm{s}(\alpha\gamma) = O(\mathrm{s}(\alpha) + \mathrm{s}(\gamma) + mF) = O\big(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2F\big),$$
$$\mathrm{s}(n) = O\big(m\,\mathrm{s}(\beta) + m^2F\big).$$

Ezeket behelyettesítve (2.2.27)-be és hozzáadva $\mathrm{T}(\alpha/\beta)$-t megkapjuk (2.2.28)-at. ■

2.2.3. Műveletek összefoglalása

A következő táblázatban összefoglaljuk a 2.2. szakasz eredményeit a $\mathbb{Z}[\theta]$-beli műveletek futási idejéről és az eredmények méretéről. Legyen $\alpha, \beta, \gamma \in \mathbb{Z}[\theta]$ és $n \in \mathbb{Z}$.

2.2.17. Táblázat: Műveletek hatékonysága $\mathbb{Z}[\theta]$-ban

Művelet	Eredmény mérete	Időkorlát
$\alpha \pm \beta$	$\mathrm{s}(\alpha, \beta) + \log 2$	$O(m\,\mathrm{s}(\alpha, \beta))$
$n\alpha$	$\mathrm{s}(\alpha) + \mathrm{s}(n)$	$O(m\,\mathrm{Mul}(\mathrm{s}(n), \mathrm{s}(\alpha)))$
$\alpha\beta$	$O(\mathrm{s}(\alpha) + \mathrm{s}(\beta) + mF)$	$O\big(m^2\,\mathrm{Mul}(\mathrm{s}(\alpha), \mathrm{s}(\beta)) + m^2\,\mathrm{Mul}(\mathrm{s}(\alpha) + \mathrm{s}(\beta) + \log m,\, mF)\big)$
$\alpha^{-1} \to \frac{\beta}{n}$	$O(m\,\mathrm{s}(\alpha) + m^2F)$	$O(m^3\,\mathrm{Mul}(m\,\mathrm{s}(\alpha) + m^2F))$
$\frac{\alpha}{\beta} \to \frac{\gamma}{n}$	$O(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2F)$	$O(m^3\,\mathrm{Mul}(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2F))$
$\alpha < \beta,\ \alpha \leq \beta$	$O(1)$	$O(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha, \beta) + m^2F))$
$\lfloor \frac{\alpha}{n} \rfloor, \lceil \frac{\alpha}{n} \rceil, \lfloor \frac{\alpha}{n} \rceil$	$O(\mathrm{s}(\alpha) + mF)$	$O(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha, n) + m^2F))$
$\lfloor \frac{\alpha}{\beta} \rfloor, \lceil \frac{\alpha}{\beta} \rceil, \lfloor \frac{\alpha}{\beta} \rceil$	$O(\mathrm{s}(\alpha) + m\,\mathrm{s}(\beta) + m^2F)$	$O(m^2\,\mathrm{Mul}(m\,\mathrm{s}(\alpha) + m^2\,\mathrm{s}(\beta) + m^3F))$

2.3. Bareiss-algoritmus

Lineáris egyenletrendszerek megoldására közismert eszköz a Gauss-elimináció. Ha az alaphalmaz $\mathbb{Z}$ vagy $\mathbb{Q}$, és egzaktul szeretnénk számolni, akkor a Gauss-elimináció során a számok hossza exponenciálisan növekedhet, vagy a racionális számok egyszerűsítése lelassítja az algoritmust. Ennek a megoldására készült a Bareiss-algoritmus [7], amely ügyesen beiktatott egzakt osztásokkal polinomiális korlát alatt tartja a számok méretét.

Ebben a szakaszban kiterjesztjük az algoritmust $\mathbb{Z}[\theta]$-ra (ahol θ most már nem feltétlenül valós), és kiszámítjuk a futási idejét az előző szakasz eredményeire támaszkodva. A Bareiss-algoritmus legegyszerűbb formáját vizsgáljuk, amikor egy négyzetes mátrixot felső háromszög alakra hozunk (például hogy kiszámítsuk a determinánsát).

Legyen $M \in \mathbb{Z}[\theta]^{n\times n}$, melynek elemei α_{ij}. A Bareiss-algoritmus működése a következő rekurzív formulával foglalható össze [7, 570. o.]:

$$\alpha_{00}^{(0)} := 1, \quad \alpha_{ij}^{(1)} := \alpha_{ij}, \quad \alpha_{ij}^{(k+1)} := \frac{\alpha_{kk}^{(k)}\alpha_{ij}^{(k)} - \alpha_{ik}^{(k)}\alpha_{kj}^{(k)}}{\alpha_{k-1,k-1}^{(k-1)}}, \tag{2.3.1}$$

ahol $1 \leq k \leq n-1$ és $k+1 \leq i, j \leq n$.

Először kiszámítunk egy korlátot a menet közbeni $\alpha_{ij}^{(k)}$ értékek méretére az eredeti α_{ij} mátrixelemek méretének függvényében. Minden $\alpha_{ij}^{(k)}$-t felírhatunk egy α_{ij}-kből álló $k \times k$ méretű determinánsként [7, 565. o.]. Ebből következik, hogy a (2.3.1)-beli osztás valóban egzakt, és $\alpha_{ij}^{(k)} \in \mathbb{Z}[\theta]$, tehát használhatjuk $\mathrm{s}(\cdot)$-et a mérésére. Legyen $A := \max_{i,j} \mathrm{s}(\alpha_{ij})$ és $B := \max_{i,j,k} \mathrm{s}\left(\alpha_{ij}^{(k)}\right)$, ekkor a 2.2.7. Lemma alapján:

$$B = O(nA + nmF + n\log n).$$

Most megvizsgáljuk a (2.3.1) rekurzív formula futási idejét. Az alábbiakban felsoroljuk a főbb lépéseit, és feltüntetjük azok futási idejét és az eredményének méretét s(·)-ben (mindegyik könnyen kiszámítható a 2.2.17. Táblázat alapján):

1. két szorzás:
   - idő: $O(m^2\,\mathrm{Mul}(B))$ (2.2.19),
   - méret: $O(B)$ (2.2.11);
2. egzakt osztás:
   (a) kiszámítjuk $\left(\alpha_{k-1,k-1}^{(k-1)}\right)^{-1}$-et $\beta_{k-1,k-1}^{(k-1)}/d_{k-1,k-1}^{(k-1)}$ formában:
   - idő: $O(m^3\,\mathrm{Mul}(mB))$ (2.2.22),
   - méret: $O(mB)$ (2.2.12), (2.2.13);

   (b) megszorozzuk a számlálót $\beta_{k-1,k-1}^{(k-1)}$-gyel:
   - idő: $O(m^2\,\mathrm{Mul}(B, mB)) = O(m^3\,\mathrm{Mul}(B))$ (2.2.19),
   - méret: $O(mB)$ (2.2.11);

   (c) végül elosztjuk az eredményül kapott algebrai egészet $d_{k-1,k-1}^{(k-1)}$-gyel egzaktul:
   - idő: $O(m\,\mathrm{Mul}(mB))$.

A fenti lépések többségét minden i, j, k-ra el kell végezni, azaz $O(n^3)$-ször, kivéve 2/(a)-t, amely csak k-tól függ, így azt $O(n)$-szer végezzük el. Ezeket mind összeadva:

$$\mathrm{T}_{\mathbb{Z}[\theta]}(\mathrm{Bareiss}) = O\big(n^3 m^3\,\mathrm{Mul}(B) + n\,m^3\,\mathrm{Mul}(mB)\big). \tag{2.3.2}$$

Összehasonlításként a futási idő $\mathbb{Z}$ fölött:

$$\mathrm{T}_{\mathbb{Z}}(\mathrm{Bareiss}) = O\big(n^3\,\mathrm{Mul}(nA + n\log n))\big). \tag{2.3.3}$$

Látható, hogy ha eltekintünk a θ-tól függő konstansoktól (m és F), akkor a két futási idő nagyságrendileg azonos. A következő táblázatban összehasonlítjuk az eredményt különböző Mul(·) függvényekre (lásd a 2.2.2. szakasz elejét) mindkét esetben. Az utolsó sorban bevezettünk egy egyszerűsített jelölést a logaritmikus faktorok kiküszöbölésére: $\widetilde{O}(N) := O(N\,\log N\,\log\log N)$.

2.3.1. Táblázat: A Bareiss-algoritmus futási ideje

$\mathrm{Mul}(X)$	$\mathrm{T}_{\mathbb{Z}[\theta]}(\mathrm{Bareiss})$	$\mathrm{T}_{\mathbb{Z}}(\mathrm{Bareiss})$
X^2	$O(n^3m^3(n^2+m^2)(A+mF+\log n)^2)$	$O(n^5(A+\log n)^2)$
$X^{\log_2 3}$	$O(n^{2,6}m^3(n^2+m^{1,6})(A+mF+\log n)^{1,6})$	$O(n^{4,6}(A+\log n)^{1,6}))$
$X\log X\log\log X$	$\widetilde{O}(n^2m^3(n^2+m)(A+mF+\log n))$	$\widetilde{O}(n^4(A+\log n))$

2.4. LLL-algoritmus

Az LLL-algoritmus egy rácsredukciós algoritmus, amely egy rács egy bázisát ugyanannak a rácsnak egy redukált bázisává alakítja (ezt rövidesen precízen definiáljuk). Az algoritmust Lenstra, Lenstra és Lovász írták meg [25]-ben, és kielemezték a hatékonyságát egész vektorokra ($\mathbb{Z}^n$-ben). Azóta többen vizsgálták $\mathbb{R}^n$-ben is (numerikusan), például [26, 28, 30, 32]. Mi ehelyett azzal foglalkozunk, hogy mi történik, ha valós algebrai számokkal számolunk egzaktul $\mathbb{Z}[\theta]^n$-ben. Néhányan már alkalmazták az algoritmust algebrai számtestekre (pl.: [17, 22, 29]), de nem tudunk arról, hogy az egzakt számolás hatékonyságával bárki foglalkozott volna.

Felmerülhet a kérdés, hogy mi értelme van az LLL-algoritmust egzaktul számolni, hiszen a legtöbb alkalmazásban nincs szükség az egzakt eredményre, csak egy „eléggé redukált" bázisra vagy egy elég rövid vektorra, és a numerikus számolás sokkal gyorsabb. Azért foglalkozunk ezzel mégis, mert egyrészt elméleti szempontból érdekes kérdés egy ismert algoritmust más szemszögből vizsgálni, másrészt vannak olyan alkalmazások is, ahol kifejezetten az egzakt eredményre van szükség. Ilyen például [29], amely algebrai egészekhez rendel egész vektorokból álló periodikus sorozatot, amelyet az LLL-algoritmus ismételt alkalmazásával számít ki. Ez bizonyos értelemben a lánctörtkifejtés általánosítása, és ahhoz hasonlóan csak egzakt számolással garantálható helyes végeredmény.

Az LLL-algoritmus bemenetként kap egy $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ bázist, és lépésenként módosítja a vektorokat addig, amíg bizonyos (rövidesen definiált) tulajdonságokat nem teljesítenek. Az algoritmus végig megőrzi a vektorok által kifeszített rácsot:

2.4.1. Definíció. Ha $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$, akkor

$$\Lambda(b_1, b_2, \ldots, b_n) := \left\{ c_1 b_1 + c_2 b_2 + \ldots + c_n b_n \,\middle|\, c_1, c_2, \ldots, c_n \in \mathbb{Z} \right\}$$

a $b_1, b_2, \ldots, b_n$ vektorok által kifeszített *rács*.

Az algoritmushoz szükségünk lesz a következő konstrukcióra:

2.4.2. Definíció. A $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ vektorrendszer *Gram–Schmidt-ortogonalizált*ja a $b_1^*, b_2^*, \ldots, b_n^*$ vektorrendszer, amelyet a következő rekurzív formulákkal definiálunk:

$$b_i^* := b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^* \qquad (1 \le i \le n), \tag{2.4.1}$$

$$\mu_{ij} := \frac{\langle b_i, b_j^* \rangle}{\langle b_j^*, b_j^* \rangle} \qquad (1 \le j < i \le n), \tag{2.4.2}$$

ahol $\langle \cdot, \cdot \rangle$ a szokásos skaláris szorzás: $\langle x, y \rangle := x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$.

Az algoritmus bármelyik pillanatában szükségünk lesz az aktuális vektorok Gram–Schmidt-ortogonalizáltjára, így a továbbiakban b_i^* és μ_{ij} automatikusan ezeket jelöli az aktuális b_i-kre.

Az algoritmus befejezésekor a vektorok a következő feltételt teljesítik:

2.4.3. Definíció. A $b_1, b_2, \ldots, b_n \in \mathbb{R}^n$ bázis *LLL-redukált*, ha

$$|\mu_{ij}| \leq \frac{1}{2} \qquad (1 \leq j < i \leq n), \tag{2.4.3}$$

$$\left\|b_i^* + \mu_{i\,i-1} b_{i-1}^*\right\|_2^2 \geq \delta \left\|b_{i-1}^*\right\|_2^2 \qquad (2 \leq i \leq n), \tag{2.4.4}$$

ahol δ az algoritmus paramétere, $\frac{1}{4} < \delta < 1$, gyakran $\delta = \frac{3}{4}$, és $\|\cdot\|_2$ a szokásos euklideszi norma: $\|x\|_2 := \sqrt{\langle x, x\rangle}$.

Az LLL-algoritmus váza a következő. Itt csak a b_i-k módosításait mutatjuk be, de a teljes algoritmus frissíti a b_i^* és a μ_{ij} értékeket is (vagy azzal ekvivalens változókat), hogy azok mindig teljesítsék a 2.4.2. Definíciót.

2.4.4. Algoritmus: LLL-algoritmus (vázlat)

```
k := 2
while k ≤ n do
    if k ≥ 2 then
        b_k := b_k − ⌊μ_{k k−1}⌉ b_{k−1}
    if k ≥ 2 ∧ ‖b*_k + μ_{k k−1} b*_{k−1}‖_2^2 < δ‖b*_{k−1}‖_2^2 then
        (cserélő lépés)
        b_k ↔ b_{k−1}
        k := k − 1
    else
        (redukciós lépés)
        for l := k − 2 to 1 do
            b_k := b_k − ⌊μ_{kl}⌉ b_l
        k := k + 1
```

Az algoritmus elemzésekor már az sem triviális kérdés, hogy a fő **while**-ciklus hány lépést tesz meg, vagy hogy egyáltalán megáll-e valaha. A [25] szerzői bebizonyították, hogy az algoritmus véges lépésben terminál, és $\mathbb{Z}^n$-re kiszámítottak egy korlátot a futási időre. A továbbiakban ezt fogjuk általánosítani $\mathbb{Z}[\theta]^n$-re. Ez egyáltalán nem könnyű feladat, mert [25]-ben sokszor kihasználják $\mathbb{Z}$-nek azt a tulajdonságát, hogy egy pozitív szám biztosan legalább 1, viszont $\mathbb{Z}[\theta]$-ban ez nem teljesül, sőt, a pozitív számok tetszőlegesen közel lehetnek a 0-hoz.

Az LLL-algoritmus futási idejét három összetevőből rakjuk össze: adunk egy becslést a **while**-ciklus lépéseinek számára ($\mathbb{Z}^n$-nél általánosabban), kiszámítunk egy korlátot az algoritmus változóinak menet közbeni méretére $\mathbb{Z}[\theta]$-ban, és ennek segítségével megbecsüljük az egyes lépések futási idejét.

2.4.1. Tulajdonságok $\mathbb{R}^n$-ben

Mielőtt rátérnénk az algebrai számokra, először megvizsgáljuk az LLL-algoritmus néhány tulajdonságát általában, $\mathbb{R}^n$-ben. Kezdjük néhány jelöléssel.

Szükségünk lesz egy adott $\Lambda := \Lambda(b_1, b_2, \ldots, b_n)$ rácsban a legrövidebb vektor hosszára:

$$L_0 := \min\bigl\{\, \|x\|_2^2 \;\bigm|\; x \in \Lambda \setminus \{0\} \bigr\}.$$

További fontos mennyiségek a $d_1, d_2, \ldots, d_n \in \mathbb{R}$ számok, amelyek a b_i vektoroktól függnek, és a következő három ekvivalens állítás bármelyikével definiálhatók [25, 521. o.]:

$$d_l = \|b_1^*\|_2^2 \, \|b_2^*\|_2^2 \ldots \|b_l^*\|_2^2, \tag{2.4.5}$$

$$d_l = \det(\langle b_i, b_j \rangle)_{1 \le i,j \le l}, \tag{2.4.6}$$

$$d_l = \det(\Lambda(b_1, b_2, \ldots, b_l))^2. \tag{2.4.7}$$

Célszerű kiterjeszteni a jelölést $d_0 := 1$-re is. Ezeknek a számoknak az a jelentősége, hogy egyrészt (2.4.5) miatt szoros kapcsolatban állnak a b_i^* vektorokkal, így azok helyett használhatók az algoritmusban (majd később látni fogjuk, pontosan hogyan), másrészt (2.4.6) miatt törtmentesek, azaz ha $b_i \in \mathbb{Z}^n$, akkor $d_l \in \mathbb{Z}$, tehát könnyebb velük számolni. Látni fogjuk, hogy ez utóbbi tulajdonságot algebrai számokra is ki tudjuk terjeszteni.

Ezen mennyiségekkel kapcsolatban szükségünk lesz a következő egyenlőtlenségekre, amelyek függetlenek az LLL-algoritmus kontextusától (tehát annak bármelyik pillanatában érvényesek):

2.4.5. Lemma.

$$d_l \ge \left(\frac{L_0}{l}\right)^l \qquad (1 \le l \le n), \tag{2.4.8}$$

$$\min_{i=1}^{n} \|b_i^*\|_2^2 \le L_0 \le \|b_1^*\|_2^2. \tag{2.4.9}$$

Bizonyítás. Minkowski tétele [13, III.2.2.] szerint ha $S \subseteq \mathbb{R}^l$ egy konvex, origóra szimmetrikus mérhető alakzat, és a $\Lambda_l := \Lambda(b_1, b_2, \ldots, b_l)$ ráccsal $S \cap \Lambda_l = \{0\}$, akkor

$$\det(\Lambda_l) \ge 2^{-l} \operatorname{Vol}(S),$$

ahol $\operatorname{Vol}(S)$ az S térfogata (l dimenzióban). Alkalmazzuk ezt a tételt az l-dimenziós

kockára, amelynek oldala $2r/\sqrt{l}$ (így a körülírt gömbjének sugara r), ahol $r < \sqrt{L_0}$. Ekkor ha $r \to \sqrt{L_0}$, akkor négyzetre emelés után – (2.4.7) miatt – megkapjuk (2.4.8)-at.

A (2.4.9) második része a (2.4.8) speciális esete $l = 1$-re, mivel $d_1 = \|b_1^*\|_2^2$. Az állítás első fele elemi, és következik például a [25, (1.11)] bizonyításából. ■

A következő lemmában megmutatjuk, hogy az LLL-algoritmus változóit és egyéb mennyiségeit akárhány (egész számú) lépés után megbecsülhetjük a kezdeti adatok segítségével.

2.4.6. Lemma. *Legyen $B := \max_{i=1}^n \|b_i\|_2^2$ az LLL-algoritmus bemenő vektoraira. Ekkor valahányszor a fő* **while**-*ciklus törzsének elején vagy végén tart az algoritmus, a következő egyenlőtlenségek teljesülnek (a k ciklusváltozó függvényében):*

$$\|b_i^*\|_2^2 \le B, \tag{2.4.10}$$

$$\|b_i\|_2^2 \le nB \qquad (i \ne k), \tag{2.4.11}$$

$$|\mu_{ij}| \le \frac{1}{2} \qquad (i < k), \tag{2.4.12}$$

$$|\mu_{ij}| \le 2^{n-i}\sqrt{n}\left(\frac{nB}{L_0}\right)^{\frac{n-1}{2}} \qquad (i = k), \tag{2.4.13}$$

$$|\mu_{ij}| \le \sqrt{n}\left(\frac{jB}{L_0}\right)^{\frac{j}{2}} \qquad (i > k), \tag{2.4.14}$$

$$d_j \le B^j. \tag{2.4.15}$$

Bizonyítás. Ezek az egyenlőtlenségek hasonlítanak a [25, 523. o.]-on lévőkre, és közülük (2.4.10), (2.4.11), (2.4.12) és (2.4.15) pontosan ugyanúgy bizonyítható, mint [25]-ben. A többit nem vehetjük át közvetlenül, mert [25] $\mathbb{Z}$ fölött dolgozik, és kihasználja, hogy $d_l \ge 1$. Itt ezt lecseréljük az általánosabb (2.4.8) alsó korlátra.

A (2.4.14)-et az eddigiekből és a Cauchy–Schwarz-egyenlőtlenségből vezetjük le:

$$|\mu_{ij}|^2 \overset{(2.4.2)}{=} \frac{\left|\langle b_i, b_j^*\rangle\right|^2}{\|b_j^*\|_2^4} \overset{\text{C.-S.}}{\le} \frac{\|b_i\|_2^2\,\|b_j^*\|_2^2}{\|b_j^*\|_2^4} \overset{(2.4.5)}{=} \frac{d_{j-1}}{d_j}\|b_i\|_2^2 \overset{\substack{(2.4.15)\\(2.4.8)}}{\le} \frac{B^{j-1}}{\left(\frac{L_0}{j}\right)^j}\|b_i\|_2^2 \overset{(2.4.11)}{\le} n\left(\frac{jB}{L_0}\right)^j.$$

A (2.4.13)-at ugyanúgy vezetjük le (2.4.14)-ből, mint [25, (1.34)]-et [25, (1.33)]-ból. ■

Az eddigiekből már becslést tudunk adni az LLL-algoritmus lépéseinek számára.

2.4.7. Lemma. *Legyen N az LLL-algoritmus fő lépéseinek száma (a redukciós és a cserélő lépések összesen), és legyen $K_\delta := \frac{1}{\log\frac{1}{\delta}}$. Ekkor*

$$N = O\left(n^2 \log \frac{nB}{L_0} K_\delta\right).$$

Bizonyítás. A bizonyítás [25]-ön alapul, de az kihasználja, hogy a vektorok $\mathbb{Z}^n$-ből vannak, így azt közvetlenül nem tudjuk alkalmazni.

Legyen N_r a redukciós lépések száma, és N_s a cserélő lépéseké. Mivel az előbbi növeli, az utóbbi pedig csökkenti k-t 1-gyel, és mivel az algoritmus $k = 2$-vel indul és $k = n+1$-gyel áll meg, ezért $N_r - N_s = n - 1$, tehát $N = N_r + N_s = 2N_s + n - 1$, vagyis elegendő N_s meghatározása.

Legyen $D := d_1 d_2 \dots d_n$, és legyen $D^{(s)}$ a D értéke pontosan s cserélő lépés után. Becsüljük D-t két oldalról:

$$D = \prod_{j=1}^{n} d_j \overset{(2.4.8)}{\geq} \prod_{j=1}^{n} \left(\frac{L_0}{j}\right)^j \geq \prod_{j=1}^{n} \left(\frac{L_0}{n}\right)^j = \left(\frac{L_0}{n}\right)^{\frac{n(n+1)}{2}},$$

$$D = \prod_{j=1}^{n} d_j \overset{(2.4.15)}{\leq} \prod_{j=1}^{n} B^j = B^{\frac{n(n+1)}{2}}.$$

Ezek a korlátok akárhány lépés után is érvényesek, azaz bármelyik $D^{(s)}$-re is. [25, 521. o.] bebizonyítja – $\mathbb{Z}$ specialitásának használata nélkül –, hogy a redukciós lépés nem változtatja D-t, a cserélő lépés pedig csökkenti D-t egy δ-nál kisebb szorzóval: $D^{(s+1)} < \delta D^{(s)}$. Ebből indukcióval következik, hogy $D^{(s)} < \delta^s D^{(0)}$. Az egyenlőtlenségeket összeillesztve:

$$\left(\frac{L_0}{n}\right)^{\frac{n(n+1)}{2}} \leq D^{(N_s)} < \delta^{N_s} D^{(0)} \leq \delta^{N_s} B^{\frac{n(n+1)}{2}}.$$

Vegyünk mindkét végéből logaritmust, és rendezzük át a következőképpen:

$$N_s < \frac{1}{\log \frac{1}{\delta}} \frac{n(n+1)}{2} \log \frac{nB}{L_0},$$

ebből pedig már következik az állítás, mert fent láttuk, hogy $N = 2N_s + n - 1$. ■

Megjegyezzük, hogy $\mathbb{Z}^n$-ben egyszerűbb a bizonyítás, mert akkor D egész szám, így $D \geq 1$. Ezt kihasználva [25] bebizonyítja, hogy a lépések száma $O(n^2 \log B)$. A fenti bizonyításból is könnyen ellenőrizhetjük ezt, ha lecseréljük D alsó korlátját 1-re: ekkor $O(n^2 \log B\, K_\delta)$ jön ki, amely a δ-tól való függést is feltünteti.

2.4.2. Változók mérete $\mathbb{Z}[\theta]^n$-ben

Mostantól áttérünk algebrai számtestekre. Használjuk a 2.1.2. szakaszban és (2.2.1)-ben bevezetett jelöléseket: θ, m, F, $\mathrm{s}(\cdot)$, valamint kiterjesztjük $\mathrm{s}(\cdot)$-et vektorokra is: $\mathrm{s}(x) := \max_{j=1}^{n} \mathrm{s}(x_j)$. Itt speciálisan valós algebrai számokkal számolunk: $\theta \in \mathbb{R}$, és a bemeneti vektorok $b_1, b_2, \dots, b_n \in \mathbb{Z}[\theta]^n \subset \mathbb{R}^n$. (Ha $b_i \in K^n = \mathbb{Q}(\theta)^n$, akkor előre

felszorozhatunk a közös nevezővel, és $\mathbb{Z}[\theta]^n$-be jutunk.) A vektorok együtthatóméretére vezessük be a következő jelölést:

$$A := \max_{i=1}^{n} \mathrm{s}(b_i). \tag{2.4.16}$$

Ha az LLL-algoritmust (2.4.4. Algoritmus) egzaktul (nem lebegőpontosan) szeretnénk implementálni, akkor az a probléma, hogy b_i^* és μ_{ij} törtek is lehetnek, és ez megnehezíti a számítást. Egész vektorokra azonban van egy ezzel ekvivalens módosított algoritmus: [14, Alg. 2.6.7], amely csak egész változókat használ: b_i^* és μ_{ij} helyett $d_j \in \mathbb{Z}$-t, valamint a $\mu_{ij} = \lambda_{ij}/d_j$ felírásból λ_{ij}-t, amely szintén egész (lásd: [14, Prop. 2.6.5]). Könnyen belátható, hogy ez a módosított változat nemcsak $\mathbb{Z}$-re, hanem $\mathbb{Z}[\theta]$-ra is működik, azaz $d_j, \lambda_{ij} \in \mathbb{Z}[\theta]$. Ezt részletesen a következő, 2.4.3. szakaszban fogjuk bemutatni.

Ebben a szakaszban korlátokat adunk $\mathrm{s}(d_j)$-re és $\mathrm{s}(\lambda_{ij})$-re. Első lépésként ezt az aktuális $\mathrm{s}(b_i)$ segítségével tesszük meg, függetlenül az LLL-algoritmus kontextusától.

2.4.8. Lemma. *d_j és λ_{ij} a következőképpen becsülhető a megfelelő b_i-k segítségével:*

$$\mathrm{s}(d_j, \lambda_{ij}) \leq 2n\Big(\max_{i=1}^{n} \mathrm{s}(b_i) + (m-1)F + 2\log m + \log n\Big).$$

Bizonyítás. (2.4.6)-ból tudjuk, hogy d_j felírható $j \times j$ méretű determinánsként $\langle b_{i'}, b_{j'}\rangle$-típusú elemekből. Azt állítjuk, hogy λ_{ij} is hasonló struktúrájú. Ez következik például [14, Prop. 2.6.5] bizonyításából, amely azt mutatja meg, hogy $\lambda_{ij} \in \mathbb{Z}$, és közben kihozza (nem használva $\mathbb{Z}$ tulajdonságait), hogy

$$\begin{pmatrix} \langle b_1, b_1\rangle & \cdots & \langle b_1, b_j\rangle \\ \vdots & \ddots & \vdots \\ \langle b_j, b_1\rangle & \cdots & \langle b_j, b_j\rangle \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_j \end{pmatrix} = \begin{pmatrix} \langle b_1, b_i\rangle \\ \vdots \\ \langle b_j, b_i\rangle \end{pmatrix},$$

ahol $\xi_j = \mu_{ij}$. Ha ezt megoldjuk a Cramer-szabállyal $\mu_{ij} = \lambda_{ij}/d_j$-re, akkor éppen a kívánt típusú determinánsokat kapjuk λ_{ij}-re és d_j-re.

Szükségünk van a $\langle b_{i'}, b_{j'}\rangle$-típusú elemek becslésére. Az $\mathrm{s}(\cdot)$ operátor tulajdonságait ((2.2.9) és (2.2.11)) felhasználva könnyen belátható, hogy

$$\mathrm{s}(\langle b_{i'}, b_{j'}\rangle) \leq \mathrm{s}(b_{i'}) + \mathrm{s}(b_{j'}) + (m-1)F + 2\log m + \log n,$$

és alkalmazzuk a 2.2.7. Lemmát d_j-re és λ_{ij}-re, mint $j \times j$ méretű determinánsokra. ■

Az előző szakaszban, ahol az LLL-algoritmus általános ($\mathbb{R}$ fölötti) tulajdonságait tárgyaltuk, számos eredményben szerepelt az L_0 konstans, legtöbbször $\frac{nB}{L_0}$ formában. $\mathbb{Z}[\theta]$-ban megszabadulhatunk L_0-tól a következő lemma segítségével, és minden eredményt felírhatunk az alapvető paraméterekkel (n, m, A, F, δ).

2.4.9. Lemma. *Tekintsük az LLL-algoritmust $\mathbb{Z}[\theta]$ fölött, és legyen*

$$H := \frac{1}{n} \log \frac{nB}{L_0}, \tag{2.4.17}$$

ekkor

$$H = O\big(mA + m^2F + m \log n\big). \tag{2.4.18}$$

Bizonyítás. L_0-ra a következő alsó korlátot adhatjuk a kezdeti d_l-ek segítségével:

$$\frac{1}{L_0} \overset{(2.4.9)}{\leq} \max_{i=1}^{n} \frac{1}{\|b_i^*\|_2^2} \overset{(2.4.5)}{=} \max_{i=1}^{n} \frac{d_{i-1}}{d_i}.$$

Alkalmazzuk a (2.2.14) és (2.2.15) egyenlőtlenségeket d_l becslésére két oldalról, és használjuk a 2.4.8. Lemmát $\mathrm{s}(d_l)$ becslésére:

$$\log \frac{d_{i-1}}{d_i} = \log d_{i-1} + \log d_i^{-1} < m \max_{j=0}^{n} \mathrm{s}(d_j) + O(m^2F) = O\big(nmA + nm^2F + nm \log n\big).$$

Adnunk kell továbbá egy felső becslést B-re, amihez szintén (2.2.14)-et használjuk:

$$\log B = \log \max_{i=1}^{n} \|b_i\|_2^2 \leq 2 \log \max_{i=1}^{n} \|b_i\|_\infty + \log n < 2A + 2mF + \log n.$$

A bizonyítandó állítást megkapjuk ezen egyenlőtlenségek kombinálásával. ■

2.4.10. Lemma. *Tekintsük az LLL-algoritmus egy fő lépését (egy redukciós lépést vagy egy cserélő lépést). Jelöljük a vektorokat a lépés előtt b_i-vel, utána pedig b_i'-vel, ekkor*

$$\mathrm{s}(b_i') \leq \max_{l=1}^{n} \mathrm{s}(b_l) + \frac{n^2}{2} H + n \log 2.$$

Bizonyítás. Az alábbi algoritmusrészlet bemutatja b_i és μ_{ij} változásait egy redukciós lépés során. (Megjegyzés: a 2.4.4. Algoritmushoz képest itt szerepelnek a μ_{kj}-k is, és a b_k összes változását összegyűjtöttük az l-ciklusba. Azt is jegyezzük meg, hogy ezt az algoritmusrészletet a számítás kedvéért mutatjuk be, de valójában nem így, hanem a d_j és a λ_{ij} segítségével számolunk, lásd a 2.4.3. szakaszban.)

2.4.11. Algoritmus: Redukciós lépés (az LLL-algoritmusban)

```
for l := k − 1 to 1 do
    b_k := b_k − ⌊μ_kl⌉ b_l
    for j := 1 to l − 1 do
        μ_kj := μ_kj − ⌊μ_kl⌉ μ_lj
    μ_kl := μ_kl − ⌊μ_kl⌉
```

A cserélő lépés ezekből csak az $l = k - 1$ lépést teszi meg, valamint felcserél még két b_i-t, de az nem változtatja meg az $\mathrm{s}(b_i)$-k maximumát, így elegendő csak a redukciós lépéssel foglalkozni.

Ha $i \neq k$, akkor b_i és μ_{ij} nem változik, tehát az állítás triviálisan teljesül ezekre a b_i-kre. Az $\mathrm{s}(b_k)$-k változásának kiszámításához szükségünk van $\lfloor \mu_{kl} \rceil$ becslésére. Megkülönböztetésül jelölje μ_{kl} a kezdeti értéket, és μ'_{kl} a $\lfloor \cdot \rceil$ kiszámításakor érvényes értéket. A fenti algoritmusrészletből könnyen levezethető, hogy

$$\mu'_{kl} = \mu_{kl} - \sum_{i=l+1}^{k-1} \lfloor \mu'_{ki} \rceil \mu_{il},$$

így a μ_{ij}-re vonatkozó (2.4.12) és (2.4.13) egyenlőtlenségek felhasználásával:

$$\left|\lfloor \mu'_{kl} \rceil\right| \leq 2|\mu'_{kl}| \leq 2|\mu_{kl}| + 2\sum_{i=l+1}^{k-1} \left|\lfloor \mu'_{ki} \rceil\right| |\mu_{il}| \leq 2^{n-k+1}\sqrt{n}\left(\frac{nB}{L_0}\right)^{\frac{n-1}{2}} + \sum_{i=l+1}^{k-1} \left|\lfloor \mu'_{ki} \rceil\right|.$$

Ebből $l = k-1$-től $l = 1$-ig terjedő indukcióval könnyen beláthatjuk, hogy

$$\left|\lfloor \mu'_{kl} \rceil\right| \leq 2^{n-l}\sqrt{n}\left(\frac{nB}{L_0}\right)^{\frac{n-1}{2}}.$$

Most már ki tudjuk számítani az $\mathrm{s}(b_k)$ változását:

$$\mathrm{s}(b'_k) = \mathrm{s}\left(b_k - \sum_{l=1}^{k-1} \lfloor \mu'_{kl} \rceil b_l\right) \leq \max_{l=1}^{k} \mathrm{s}(b_l) + \log\left(1 + \sum_{l=1}^{k-1} \left|\lfloor \mu'_{kl} \rceil\right|\right).$$

A logaritmus argumentumára a következő felső becslést tudjuk adni a fentiekből:

$$1 + \sum_{l=1}^{k-1} \left|\lfloor \mu'_{kl} \rceil\right| \leq 1 + \sum_{l=1}^{k-1} 2^{n-l}\sqrt{n}\left(\frac{nB}{L_0}\right)^{\frac{n-1}{2}} =$$

$$= 1 + \left(2^n - 2^{n-k+1}\right) n^{\frac{1}{2}} \left(\frac{nB}{L_0}\right)^{\frac{n-1}{2}} \leq 2^n \left(\frac{nB}{L_0}\right)^{\frac{n}{2}}.$$

Az utolsó lépésben felhasználtuk, hogy $n \leq \frac{nB}{L_0}$, másképpen $L_0 \leq B$, amely következik (2.4.9)-ből és (2.4.10)-ből. A jobb oldalon kapott kifejezés logaritmusa $\frac{n^2}{2}H + n\log 2$ (lásd a H definícióját: (2.4.17)), és ezt visszahelyettesítve $\mathrm{s}(b'_k)$ fenti becslésébe megkapjuk a bizonyítandó állítást. ■

A fenti eredmények kombinálásával kiszámíthatjuk az LLL-algoritmus fő változóinak méretét menet közben:

2.4.12. Lemma. *Az LLL-algoritmusban, a fő* **while**-*ciklus törzsének elején és végén a változók a következő nagyságrendűek:*

$$\mathrm{s}(b_i) = O\big(n^5 H^2 K_\delta\big), \tag{2.4.19}$$

$$\mathrm{s}(d_j) = O\big(n^6 H^2 K_\delta\big), \tag{2.4.20}$$

$$\mathrm{s}(\lambda_{ij}) = O\big(n^6 H^2 K_\delta\big). \tag{2.4.21}$$

Bizonyítás. A 2.4.10. Lemmából indukcióval következik, hogy az LLL-algoritmus első t lépésében:

$$\mathrm{s}(b_i) \leq A + O\big(n^2 H\big)t.$$

A 2.4.7. Lemma miatt $t = O(n^3 H K_\delta)$, ezt behelyettesítve adódik (2.4.19) (mert H dominálja A-t a 2.4.9. Lemma alapján). A másik két állítás ebből és a 2.4.8. Lemmából következik. ■

2.4.3. Az algoritmus futási ideje $\mathbb{Z}[\theta]^n$-ben

Most már elegendő információval rendelkezünk ahhoz, hogy megbecsüljük az LLL-algoritmus futási idejét $\mathbb{Z}[\theta]$ fölött. Az algorimus vázlatát a 2.4.4. Algoritmusban mutattuk be, a részleteket pedig [14, Alg. 2.6.7]-ből vesszük, de $\mathbb{Z}$-ről $\mathbb{Z}[\theta]$-ra általánosítva.

Először kiszámítjuk az algoritmus következő építőköveinek futási idejét külön-külön:

1. T(red): egy redukció μ_{kl}-lel, azaz a $b_k := b_k - \lfloor \mu_{kl} \rceil b_l$ értékadás, és a többi változó ennek megfelelő frissítése. (Megjegyzés: ez nem azonos egy teljes *redukciós lépés*sel, amely ezt a redukciót $k-1$-szer végzi el.)
2. T(cs): a b_k és b_{k-1} felcserélése és a többi változó ennek megfelelő frissítése.
3. T(öh): a $\big\|b_k^* + \mu_{k\,k-1} b_{k-1}^*\big\|_2^2 < \delta \big\|b_{k-1}^*\big\|_2^2$ összehasonlítás.

Legyen D az $\mathrm{s}(d_j)$, $\mathrm{s}(\lambda_{ij})$ és $\mathrm{s}(b_i)$ közös felső korlátja az algoritmus akárhány lépése után. A 2.4.12. Lemmából tudjuk, hogy $D = O(n^6 H^2 K_\delta)$.

Először nézzük T(red)-et. A redukció a következő lépésekből áll:

2.4.13. Algoritmus: A $b_k := b_k - \lfloor \mu_{kl} \rceil b_l$ redukció

$q := \left\lfloor \frac{\lambda_{kl}}{d_l} \right\rceil$
$b_k := b_k - q b_l$
for $j := 1$ **to** $l-1$ **do**
  $\lambda_{kj} := \lambda_{kj} - q\lambda_{lj}$
$\lambda_{kl} := \lambda_{kl} - q d_l$

Szükségünk van q méretének becslésére. Ez egy egész szám, és a kerekítés miatt mérete nem függ $\mathrm{s}(\lambda_{kl})$-től és $\mathrm{s}(d_l)$-től, hanem lehet jóval kisebb is:

$$\log|q| = \log\big|\lfloor \mu_{kl} \rceil\big| \leq \log 2|\mu_{kl}| \overset{(2.4.13)}{\leq} \frac{n-1}{2}\log\frac{nB}{L_0} + \frac{1}{2}\log n + n\log 2 \overset{(2.4.17)}{=} O\big(n^2 H\big).$$

A redukció főbb lépéseinek futási ideje (2.2.28) és (2.2.18) alapján a következő (kihasználjuk, hogy D dominálja mF-et, és b_l egy n-elemű vektor):

$$\mathrm{T}\left(\left\lfloor \frac{\lambda_{kl}}{d_l} \right\rceil\right) = O\big(m^2 \operatorname{Mul}(m^2 D)\big),$$
$$\mathrm{T}(q\lambda_{lj}) = \mathrm{T}(qd_l) = O\big(m \operatorname{Mul}(n^2 H, D)\big),$$
$$\mathrm{T}(qb_l) = O\big(nm \operatorname{Mul}(n^2 H, D)\big).$$

A redukció futási ideje tehát:

$$\mathrm{T}(\mathrm{red}) = \mathrm{T}\left(\left\lfloor \tfrac{\lambda_{kl}}{d_l} \right\rceil\right) + \mathrm{T}(qb_l) + (l-1)\,\mathrm{T}(q\lambda_{lj}) + \mathrm{T}(qd_l) =$$
$$= O\big(m^2 \operatorname{Mul}(m^2 D) + nm \operatorname{Mul}(n^2 H, D)\big).$$

Most számítsuk ki T(cs)-t. [14, Alg. 2.6.7] alapján a csere így végezhető el:

2.4.14. Algoritmus: A $b_k \leftrightarrow b_{k-1}$ csere

$b_k \leftrightarrow b_{k-1}$
for $j := 1$ **to** $k-2$ **do**
  $\lambda_{k,j} \leftrightarrow \lambda_{k-1,j}$
$d'_{k-1} := \frac{d_{k-2}d_k + \lambda_{k,k-1}^2}{d_{k-1}}$
for $i := k+1$ **to** n **do**
  $\lambda'_{i,k} := \frac{d_k \lambda_{i,k-1} - \lambda_{k,k-1}\lambda_{i,k}}{d_{k-1}}$
  $\lambda_{i,k-1} := \frac{d'_{k-1}\lambda_{i,k} + \lambda_{k,k-1}\lambda'_{i,k}}{d_k}$
  $\lambda_{i,k} := \lambda'_{i,k}$
$d_{k-1} := d'_{k-1}$

A futási idő szempontjából a három tört kiszámítása a kritikus művelet. Ezek hasonló struktúrájúak, mint a Bareiss-algoritmus (2.3.1) rekurzív formulája, ezért az ottanihoz nagyon hasonló számítás alkalmazható. A különbség az, hogy B helyett D van (de mindkettő dominálja mF-et), és hogy minden műveletet $O(n)$-szer végzünk el, kivéve a nevező invertálását, amelyet kétszer. Ha a 2.3. szakaszban leírthoz hasonló számítást elvégezzük, a következőt kapjuk:

$$\mathrm{T}(\mathrm{cs}) = O\big(m^3 \operatorname{Mul}(mD) + nm^3 \operatorname{Mul}(D)\big).$$

Most tekintsük a T(öh)-t. Az összehasonlítás d_j-k és λ_{ij}-k segítségével a következőképpen végezhető el:

$$d_{k-2}d_k + \lambda_{k,k-1}^2 < \delta d_{k-1}^2.$$

A szorzások (mint a csere esetén) $O(m^2 \operatorname{Mul}(D))$ ideig tartanak, az összehasonlítás ideje pedig (2.2.25) alapján $O(m^2 \operatorname{Mul}(mD))$, így összegezve:

$$\mathrm{T}(\text{öh}) = O(m^2 \operatorname{Mul}(mD)).$$

Most összeszedjük a teljes algoritmus futási idejét. A fő **while**-ciklus minden lépése vagy egy cserélő lépés, vagy egy redukciós lépés (ezeket ne tévesszük össze T(cs)-vel és T(red)-del). A cserélő lépés egy redukcióból, egy összehasonlításból és egy cseréből áll, a redukciós lépés pedig egy összehasonlításból és $k-1$ redukcióból. A kétféle lépés futási ideje tehát:

$$\mathrm{T}(\text{cs. lépés}) = \mathrm{T}(\text{öh}) + \mathrm{T}(\text{red}) + \mathrm{T}(\text{cs}) = O\big(m^2\,\mathrm{Mul}(m^2D) + nm^3\,\mathrm{Mul}(D)\big),$$

$$\mathrm{T}(\text{red. lépés}) = \mathrm{T}(\text{öh}) + (k-1)\,\mathrm{T}(\text{red}) = O\big(nm^2\,\mathrm{Mul}(m^2D) + n^2m\,\mathrm{Mul}(n^2H, D)\big).$$

Ha N a **while**-iterációk száma, akkor a 2.4.7. Lemma alapján (a 2.4.9. Lemmával kombinálva) $N = O(n^3HK_\delta)$. Ezért az LLL-algoritmus futási ideje:

$$\begin{aligned}\mathrm{T}_{\mathbb{Z}[\theta]}(\mathrm{LLL}) &\le N\,\mathrm{T}(\text{cs. lépés}) + N\,\mathrm{T}(\text{red. lépés}) = \\ &= N\,O\big(n\,m^2\,\mathrm{Mul}(m^2D) + n^2m\,\mathrm{Mul}(n^2H, D)\big) = \\ &= O\big(n^4mHK_\delta(m\,\mathrm{Mul}(n^6m^2H^2K_\delta) + n^5HK_\delta\,\mathrm{Mul}(n^2H))\big),\end{aligned}$$

ahol – ismétlésként – a jelölések a következők:

- $H = O(mA + m^2F + m\log n)$ a 2.4.9. Lemma alapján,
- n a rács dimenziója,
- m az algebrai számtest fokszáma,
- $F = \log(\max_{i=0}^m |f_i| + 1)$, ahol $f(x) = \sum_{i=0}^m f_ix^i$ ($f_m = 1$) a test primitív elemének, θ-nak a minimálpolinomja,
- $A = \max_{i=1}^n \mathrm{s}(b_i)$, a bemenő vektorok együtthatóinak mérete, és
- $K_\delta = \frac{1}{\log\frac{1}{\delta}}$, ahol δ az LLL-algoritmus paramétere ($1/4 < \delta < 1$).

Összehasonlításként idehozzuk a $\mathbb{Z}$ fölötti futási időt is. [25, (1.26) Prop.] szerint az algoritmus fő lépéseinek száma $O(n^2\log B)$, és mindegyik lépés $O(n^2)$ aritmetikai műveletet végez $O(n\log B)$ méretű egész számokon. Ha feltüntetjük a δ paramétert is, akkor a lépések száma $O(n^2\log B\,K_\delta)$ (lásd a 2.4.7. Lemma utáni megjegyzést). Ezek alapján a futási idő:

$$\mathrm{T}_{\mathbb{Z}}(\mathrm{LLL}) = O\big(n^4\log B\,K_\delta\,\mathrm{Mul}(n\log B)\big),$$

ahol $B = \max_{i=1}^n \|b_i\|_2^2$. A jobb összehasonlíthatóság kedvéért vezessük be $\mathbb{Z}^n$-ben is az $A := \log\max_{i=1}^n \|b_i\|_\infty$ jelölést, ekkor $\log B \le 2A + \log n$, és a futási idő:

$$\mathrm{T}_{\mathbb{Z}}(\mathrm{LLL}) = O\big(n^4(A + \log n)K_\delta\,\mathrm{Mul}(nA + n\log n)\big).$$

A következő táblázatban összehasonlítjuk a futási időt különböző $\mathrm{Mul}(\cdot)$ függvényekre $\mathbb{Z}[\theta]$-ban és $\mathbb{Z}$-ben:

2.4.15. Táblázat: Az LLL-algoritmus futási ideje

$\mathrm{Mul}(X)$	$\mathrm{T}_{\mathbb{Z}[\theta]}(\mathrm{LLL})$	$\mathrm{T}_{\mathbb{Z}}(\mathrm{LLL})$
X^2	$O(n^{16}m^{11}(A+mF+\log n)^5K_\delta^3)$	$O(n^6(A+\log n)^3K_\delta)$
$X^{\log_2 3}$	$O\big(n^{13,6}m^{9,4}(A+mF+\log n)^{4,2}K_\delta^{2,6}\big)$	$O(n^{5,6}(A+\log n)^{2,6}K_\delta)$
$X\log X\log\log X$	$\widetilde{O}(n^{10}(n+m^3)m^4(A+mF+\log n)^3K_\delta^2)$	$\widetilde{O}(n^5(A+\log n)^2K_\delta)$

Az utolsó sorban $\widetilde{O}(N)=O(N\log N\log\log N)$, ahogy a 2.3.1. Táblázatnál bevezettük.

2.4.4. Megjegyzések az LLL-eredményről

Bebizonyítottuk, hogy az LLL-algoritmus egzakt algebrai számok esetén sem okoz exponenciális együtthatórobbanást, és polinomiális időben végre tudjuk hajtani. Az algebrai számos időkorlát azonban lényegesen rosszabb, mint $\mathbb{Z}$ fölött, nemcsak az új paraméterek (m és F) miatt, de a többi paraméter (n és A) nagyságrendje miatt is. Ez mutatja, hogy mennyivel nehezebb az együtthatóméretet ($\mathrm{s}(\cdot)$) korlátozni, mint a közönséges méretet ($|\cdot|$), például míg $\|b_i\|_2^2\le nB$ a legtöbb i-re, addig $\mathrm{s}(b_i)=O(n^5H^2K_\delta)$.

Az eredmény azonban csak egy nagyon pesszimista elméleti felső becslés, a gyakorlatban ennél sokkal gyorsabb is lehet. Például a fő lépések száma $N=O(n^3HK_\delta)$, de ez csak egy elméleti felső korlát, a gyakorlatban gyakran néhány (pl. $O(n)$) lépés elegendő. Másik ok, hogy L_0-ra, a rács legrövidebb vektorának hosszára kiszámítottunk egy elméleti alsó korlátot: $\log\frac{1}{L_0}=O(nmA+nm^2F+nm\log n)$. A gyakorlatban azonban nincs különösebb oka, hogy miért lenne a legrövidebb vektor ennyire kicsi. Ha feltesszük, hogy ez konstans (azaz $O(1)$), akkor a futási időt jó néhány kitevővel csökkenteni tudjuk. Könnyen belátható, hogy az egyszerű szorzás ($\mathrm{Mul}(X)=X^2$) esetén ezzel a két gyakorlati feltevéssel $n^{16}m^{11}$-t le tudjuk vinni $n^{10}m^6$-ra.

A szerző azt feltételezi, hogy a kitevők a gyakorlatban tovább csökkenthetők. Ezt a jövőben gyakorlati mérésekkel szeretné alátámasztani, de ez már túlfeszíti ezen elméleti dolgozat kereteit.

3. Expanzív polinomok

3.1. Bevezetés

Ebben a fejezetben olyan algebrai számokról lesz szó, amelyek 1-nél nagyobb abszolút értékűek, és az összes konjugáltjuk (azaz a minimálpolinom többi gyöke) is ilyen tulajdonságú. Ebben az esetben a polinomot magát expanzív polinomnak nevezzük. Az elnevezés onnan ered, hogy szoros kapcsolatban állnak az expanzív mátrixokkal (ahol a sajátértékek 1-nél nagyobb abszolút értékűek), amelyek növelik a vektorok méretét bizonyos értelemben (precízebb megfogalmazásért lásd a 3.5.3. Lemmát).

3.1.1. Fogalmak és jelölések

A 3. fejezetben $f(x)$ egy n-edfokú polinomot jelöl ($n \geq 1$), amelynek együtthatói $a_0, a_1, \ldots, a_n$, gyökei pedig $\alpha_1, \ldots, \alpha_n$ ($\alpha_i \in \mathbb{C}$), azaz

$$f(x) = a_n x^n + \ldots + a_1 x + a_0 = a_n(x - \alpha_1)(x - \alpha_2)\ldots(x - \alpha_n),$$

ahol $a_n \neq 0$. Elsősorban egész együtthatós polinomokat vizsgálunk ($a_i \in \mathbb{Z}$), de egyes eredményeket általánosabban mondunk ki.

A fejezet legfontosabb fogalma a következő:

3.1.1. Definíció. Az f polinom *expanzív*, ha $\forall \alpha_i : |\alpha_i| > 1$.

Bevezetünk három mennyiséget a polinom bonyolultságára:

3.1.2. Definíció. Az f polinom

magassága (height): $$H(f) := \max_{i=0}^{n} |a_i|,$$

hosszúsága (length): $$L(f) := \sum_{i=0}^{n} |a_i|,$$

Mahler-mértéke: $$M(f) := |a_n| \prod_{i=1}^{n} \max(1, |\alpha_i|).$$

Expanzív polinomok esetén a Mahler-mérték leegyszerűsödik: $M(f) = |a_0|$. Ezeknek a fogalmaknak az a jelentősége, hogy rögzített n-re és A-ra csak véges sok olyan legfeljebb n-edfokú polinom van, amelyre $H(f) \leq A$ (vagy $L(f) \leq A$, vagy $M(f) \leq A$).

3.1.2. Korábbi eredmények

Ebben a szakaszban röviden bemutatunk néhány korábbi eredményt az expanzív polinomokkal kapcsolatban. Elsőnek lássunk néhány összefüggést az együtthatók között.

A legtriviálisabb állítás a következő:

3.1.3. Lemma. *Ha f expanzív polinom, akkor $|a_n| < |a_0|$.*

Ez egyszerűen következik abból, hogy $a_0 = \pm a_n \alpha_1 \alpha_2 \ldots \alpha_n$.

A főegyüttható és a konstans tag segítségével a közbülső együtthatókra az alábbi korlátokat tudjuk adni [10]:

3.1.4. Lemma. *Ha f expanzív, akkor*

$$|a_k| < \binom{n-1}{k-1}|a_n| + \binom{n-1}{k}|a_0| \qquad (1 \leq k \leq n-1),$$

és ez a legjobb becslés általánosan.

Ennél jobb becslést akkor lehet adni, ha több együtthatót használunk fel, például [10] adott egy becslést a_k-ra és a_{n-k}-ra ($k \leq n/2$) az $a_0, a_1, \ldots, a_{k-1}$ és $a_{n-k+1}, \ldots, a_n$ együtthatók segítségével.

A fenti feltételek szükségesek az expanzivitáshoz, de nem elégségesek. Most azt nézzük meg, hogy hogyan lehet eldönteni egy polinomról, hogy expanzív-e. Az összes gyökét kiszámítani és egyenként ellenőrizni nem célszerű, mert egyrészt nem hatékony, másrészt $n \geq 5$ fokszámú polinomokra ez csak numerikusan lehetséges, viszont szeretnénk elkerülni a közelítésekből adódó hibákat (például ha egy komplex gyök abszolút értéke hibahatáron belül 1 körül mozog).

A gyökök kiszámítása nélkül eldönthető az expanzivitás például a Schur–Cohn-teszt [15] segítségével. Ez a komplex gyökkeresésre használt Lehmer–Schur-algoritmus [23] kulcsfontosságú komponense. Először definiáljuk a Schur-transzformációt:

3.1.5. Definíció. Az f polinom *Schur-transzformáltja* a következő Tf polinom:

$$Tf(x) = b_n x^n + \ldots + b_1 x + b_0,$$

$$b_k := \overline{a_0} a_k - a_n \overline{a_{n-k}}.$$

Vegyük észre, hogy $b_n = 0$, azaz $\deg Tf \leq n-1$.

A Schur-transzformáltra a következő állítások teljesülnek:

3.1.6. Lemma. *Legyen $g := Tf$.*

1. *Ha $|a_n| < |a_0|$, akkor f-nek és g-nek ugyanannyi gyöke van az egységkörön belül.*
2. *Ha $|a_n| > |a_0|$, akkor f-nek ugyanannyi gyöke van az egységkörön kívül, mint g-nek azon belül.*
3. *Mindkét fenti esetben f és g egységkörre eső gyökei megegyeznek.*

A gyököket mindegyik esetben multiplicitással számoljuk.

Ebből az következik, hogy f akkor és csak akkor expanzív, ha $|a_n| < |a_0|$ és g expanzív.

A Schur–Cohn-teszt ezt az állítást használja rekurzívan:

3.1.7. Algoritmus: Schur–Cohn-teszt

Bemenet: f polinom
Kimenet: expanzív-e
while $n \geq 1$ **do**
  if $|a_n| \geq |a_0|$ **then**
    return false
  $f := Tf$ *(lásd: 3.1.5. Def.)*
return $f \neq 0$

A fokszám minden lépésben csökken, ezért legfeljebb n lépésben véget ér az algoritmus.

3.1.8. Példa. Legyen $f(x) := 2x^4 + x^3 - 14x^2 - 4x + 24$, ekkor a Schur–Cohn-teszt a következő polinomokat vizsgálja:

$$\begin{aligned}
f(x) &= 2x^4 + x^3 - 14x^2 - 4x + 24,\\
Tf(x) &= 32x^3 - 308x^2 - 98x + 572,\\
T^2f(x) &= -173040x^2 - 46200x + 326160,\\
T^3f(x) &= -23063040000x + 76437504000,\\
T^4f(x) &= 5310788203708416000000.
\end{aligned}$$

Mivel mindegyik polinom konstans tagja nagyobb abszolút értékű, mint a főegyütthatója (kivéve a konstans polinomot), ezért $f(x)$ expanzív.

A példából látható az algoritmus problémája: az együtthatórobbanás. A Schur-transzformált minden lépésben megduplázhatja az együtthatók hosszát, ami exponenciális együtthatónövekedéshez vezethet. Természetesen számolhatnánk lebegőpontosan, és akkor ez nem jelentene problémát, de akkor a nagy számok utolsó számjegyei elvesznének, és ez akár el is ronthatná a végeredményt.

Röviden vázolunk egy másik algoritmust is: [11]. Az ötlet az, hogy az expanzivitást egy egyszerű transzformációval visszavezetjük egy másik fogalomra, a stabilitására:

3.1.9. Definíció. Az f polinom *stabil*, ha $\forall \alpha_i : \mathrm{Re}(\alpha_i) < 0$.

3.1.10. Lemma. *Az f polinom akkor és csak akkor expanzív, ha a következő g polinom stabil:*

$$g(x) = (x+1)^n f\left(\frac{x-1}{x+1}\right) = \sum_{i=0}^{n} a_i (x+1)^{n-i}(x-1)^i.$$

A stabilitást egy egyszerű rekurzív formulával dönthetjük el:

3.1.11. Tétel. *Ha $g(x) = b_n x^n + \ldots + b_1 x + b_0$, akkor legyen*

$$c_i^{(-1)} := b_{2i}, \quad c_i^{(0)} := b_{2i+1}, \quad c_i^{(k+1)} := \frac{c_0^{(k)}}{c_0^{(k-1)}} c_{i+1}^{(k-1)} - c_{i+1}^{(k)},$$

ahol $0 \le k \le n-1$ és $0 \le i \le n/2$ (különben $c_i = 0$). Ekkor g akkor és csak akkor stabil, ha minden $c_0^{(k)}/c_0^{(k-1)} > 0$.

[11] szerint ez a módszer akkor viselkedik jobban a Schur–Cohn-tesztnél, ha a polinom valószínűleg expanzív. A probléma azonban továbbra is adott: ha egzaktul számolunk, akkor az egész számok hossza itt is minden lépésben duplázódhat. Ráadásul racionális számokat kapunk, amelyek tovább bonyolítják a számolást (pl. egyszerűsítések).

Expanzív polinomokkal kapcsolatban megemlítünk még egy problémát, amellyel itt nem foglalkozunk: rögzített fokszám és konstans tag (azaz Mahler-mérték) mellett keressük meg az összes expanzív polinomot. Erre ad algoritmust például [10, 12].

3.2. Speciális determinánsok

A 3.2.1. szakaszban szükséges és elégséges feltételeket adunk arra, hogy egy valós együtthatós polinom expanzív-e. Ehhez speciális struktúrájú determinánsokat definiálunk a polinom együtthatóiból. A 3.2.2. szakaszban megmutatjuk, hogy ha ezeket a feltételeket használjuk az expanzivitás eldöntésére, akkor ezzel elkerülhetjük az előző szakaszban bemutatott módszerek problémáját, az együtthatórobbanást, és garantáltan polinomiális időben kapunk eredményt. Ezután a 3.2.3. szakaszban általánosítjuk a bemutatott determinánsokat, és levezetjük néhány érdekes tulajdonságukat, amelyet a későbbiekben használni fogunk.

Az ebben és a következő szakaszban (3.2-3.3.) bemutatott eredmények nagyrészt a szerző következő publikációján alapulnak: [3].

3.2.1. Az expanzivitás D-feltételei

3.2.1. Definíció. Egy n-edfokú f polinomra minden $1 \le k \le n$ és mindkét előjel esetén legyen $D_k^{\pm}(f)$ egy $k \times k$ méretű determináns, amelynek az i. sorának j. eleme a következő:

$$d_{ij} = a_{j-i} \pm a_{i+j+n-k-1} \qquad (1 \le i, j \le k),$$

ahol a túlcsorduló együtthatókat nullának tekintjük, azaz $i < 0$-ra vagy $i > n$-re $a_i = 0$.

Például $n = 7$-re és $k = 6$-ra:

$$D_6^-(f) = \begin{vmatrix} a_0 - a_2 & a_1 - a_3 & a_2 - a_4 & a_3 - a_5 & a_4 - a_6 & a_5 - a_7 \\ -a_3 & a_0 - a_4 & a_1 - a_5 & a_2 - a_6 & a_3 - a_7 & a_4 \\ -a_4 & -a_5 & a_0 - a_6 & a_1 - a_7 & a_2 & a_3 \\ -a_5 & -a_6 & -a_7 & a_0 & a_1 & a_2 \\ -a_6 & -a_7 & & & a_0 & a_1 \\ -a_7 & & & & & a_0 \end{vmatrix}.$$

Ennek segítségével a következő szükséges és elégséges feltételeket tudjuk adni az expanzivitásra:

3.2.2. Tétel. *Ha f egy valós együtthatós polinom ($\forall a_k \in \mathbb{R}$) és $a_0 > 0$, akkor az alábbiak teljesülnek:*

1. *f akkor és csak akkor expanzív, ha minden $k = 1, 2, \ldots, n$-re és mindkét előjelre: $D_k^{\pm}(f) > 0$ (D-feltételek).*
2. *A $k = n$-re vonatkozó D-feltételek helyett elég a következő: $f(\pm 1) > 0$.*
3. *A D-feltételeket elég minden második k-ra feltenni, pontosabban*
   - *(a) elég $k = n$, $n-2$, $n-4$, … az (1)-es állításhoz, és*
   - *(b) elég $k = n-1$, $n-3$, $n-5$, … a (2)-es állításhoz.*

Megjegyzés: az $a_0 > 0$ feltétel nem jelent valódi megszorítást, hiszen ha $a_0 < 0$, akkor vehetjük a polinom ellentettjét, amelynek ugyanazok gyökei, ha pedig $a_0 = 0$, akkor a polinomnak gyöke a 0, tehát biztosan nem expanzív.

Bizonyítás. Felhasználjuk a Schur–Cohn-tesztet (3.1.7. Algoritmus). A bizonyításban nem tesszük fel, hogy $a_n \neq 0$, azaz megengedjük, hogy $\deg f < n$. Könnyen belátható, hogy a Schur-traszformáltról szóló 3.1.6. Lemma így is érvényes marad (feltéve, hogy $a_0 \neq 0$, de itt amúgyis feltettük, hogy $a_0 > 0$). Így technikailag könnyebben kezelhető a Schur–Cohn-teszt, mert nem kell figyelembe venni a fokszám valódi változását, elég n-et minden lépésben pontosan 1-gyel csökkenteni. Az egyértelműség kedvéért ebben a bizonyításban feltüntetjük n-et a D-feltételeken: $D_{n,k}^{\pm}$-t írunk $D_k^{\pm}$ helyett.

1. A tétel első állítását n szerinti teljes indukcióval bizonyítjuk be. A 3.1.6. Lemma alapján f expanzivitása ekvivalens azzal, hogy $|a_n| < |a_0|$, és a Schur–transzformáltja, $g := Tf$ expanzív, azaz az indukciós lépésben elég bebizonyítani, hogy

$$\begin{aligned} &\forall k \in \{1, 2, \ldots, n-1, n\} \colon D^{\pm}_{n,k}(f) > 0 \iff \\ \iff\ &|a_n| < |a_0| \wedge \forall k \in \{1, 2, \ldots, n-1\} \colon D^{\pm}_{n-1,k}(g) > 0. \end{aligned} \tag{3.2.1}$$

Ehhez elegendő két dolgot megmutatni: egyrészt hogy

$$D^{\pm}_{n,1}(f) > 0 \iff |a_n| < |a_0|, \tag{3.2.2}$$

másrészt hogy minden $k = 2, \ldots, n$-re és mindkét előjelre:

$$|a_n| < |a_0| \implies \big(D^{\pm}_{n,k}(f) > 0 \iff D^{\pm}_{n-1,k-1}(g) > 0\big). \tag{3.2.3}$$

Mivel $D^{\pm}_{n,1}(f) = a_0 \pm a_n$, és feltettük, hogy $a_0 > 0$, ezért (3.2.2) egyszerű átrendezéssel megkapható.

Most bebizonyítjuk (3.2.3)-at. Ha $n = 1$, akkor (3.2.3) üres állítás, ezért legyen $n \geq 2$.

A 3.1.5. Definíció alapján g együtthatói $b_k = a_0 a_k - a_n a_{n-k}$, így a $D^{\pm}_{n-1,k-1}(g)$ determináns elemei ($1 \leq i, j \leq k-1$):

$$\begin{aligned} d_{ij} &= b_{j-i} \pm b_{i+j+n-k-1} = \\ &= a_0(a_{j-i} \pm a_{n+i+j-k-1}) - a_n(a_{n+i-j} \pm a_{-i-j+k+1}). \end{aligned} \tag{3.2.4}$$

Azt állítjuk, hogy ez megegyezik a következő $(2k-2) \times (2k-2)$-es mátrix ($D^{(0)}$) determinánsával:

$$d^{(0)}_{ij} = \begin{cases} a_{j-i} \pm a_{n+i+j-k-1} & (1 \leq i, j \leq k-1), \\ a_{-i-j+2k} \pm a_{n+i-j+k-1} & (1 \leq i \leq k-1,\ k \leq j \leq 2k-2), \\ \pm a_n \delta_{i-k+1,j} & (k \leq i \leq 2k-2,\ 1 \leq j \leq k-1), \\ a_0 \delta_{i,j} & (k \leq i, j \leq 2k-2). \end{cases} \tag{3.2.5}$$

Ezt úgy bizonyíthatjuk be, hogy oszloptranszformációkkal kinullázzuk $D^{(0)}$ bal alsó negyedét (azaz minden $1 \leq j \leq k-1$ esetén hozzáadjuk a j. oszlophoz a $(j+k-1)$. oszlop $\mp a_n/a_0$-szorosát), ekkor könnyű látni, hogy az így kapott bal felső negyed ugyanaz, mint (3.2.4), csak megszorozva $1/a_0$-lal, a bal alsó negyed nulla, a jobb alsó negyed pedig az egységmátrix a_0-szorosa, így a teljes determináns valóban megegyezik $D^{\pm}_{n-1,k-1}(g)$-vel.

Folytatva a determináns átalakítását, vegyük észre, hogy ha $i \leq k$ és $j \geq 2$, akkor $d^{(0)}_{i,j} = d^{(0)}_{i,2k-j}$, azaz ha minden $2 \leq j \leq k-1$-re kivonjuk a j. oszlopot a $(2k-j)$-ből, akkor a következő, egyszerűbb struktúrájú $D^{(1)}$ mátrixot kapjuk:

$$d_{ij}^{(1)} = \begin{cases} a_{j-i} \pm a_{n+i+j-k-1} & (1 \le i, j \le k), \\ 0 & (1 \le i \le k,\ k+1 \le j \le 2k-2), \\ \pm a_n \delta_{i-k+1,j} & (k+1 \le i \le 2k-2,\ 1 \le j \le k), \\ a_0 \delta_{i,j} \mp a_n \delta_{i-k+1,2k-j} & (k+1 \le i, j \le 2k-2). \end{cases}$$

Ennek a determinánsa a 0-s blokk miatt két részdetermináns szorzatára bontható: a bal felső $k \times k$-s részre, amely éppen $D_{n,k}^{\pm}(f)$, és a jobb alsó $(k-2) \times (k-2)$-es részre, amely a következő:

$$\begin{vmatrix} a_0 & & & & \mp a_n \\ & a_0 & & \mp a_n & \\ & & \ddots & & \\ & \mp a_n & & a_0 & \\ \mp a_n & & & & a_0 \end{vmatrix}_{(k-2)\times(k-2)} = \begin{cases} (a_0^2 - a_n^2)^{\frac{k-2}{2}} & (2 \mid k), \\ (a_0^2 - a_n^2)^{\frac{k-3}{2}} (a_0 \mp a_n) & (2 \nmid k). \end{cases}$$

Ez a determináns pozitív, mert feltettük, hogy $|a_n| < |a_0|$. Ebből következik, hogy a másik rész, $D_{n,k}^{\pm}(f)$, valamint a teljes determináns, ami $D_{n-1,k-1}^{\pm}(g)$-vel egyenlő, azonos előjelű. Ezzel bebizonyítottuk (3.2.3)-at, és ezzel befejeztük az tétel első részének bizonyítását.

2. A tétel második állításához megmutatjuk, hogy

$$D_{n,n}^{\pm}(f) = f(\pm 1)\, D_{n,n-1}^{-}(f). \tag{3.2.6}$$

Ekkor, mivel a megmaradó D-feltételek között szerepel, hogy $D_{n,n-1}^{-}(f) > 0$, így a $D_{n,n}^{\pm}(f) > 0$ feltétel valóban lecserélhető $f(\pm 1) > 0$-ra.

A $D_{n,n}^{\pm}(f)$ mátrixán végrehajtunk egy olyan hasonlósági transzformációt, hogy az eredményen már nyilvánvaló legyen a (3.2.6) felbontás. Először bemutatjuk az ötletet egy konkrét példán $n = 4$-re: megmutatjuk, hogy $D_{4,4}^{-}(f) = f(-1)\, D_{4,3}^{-}(f)$:

$$\begin{bmatrix} 1 & -1 & 1 & -1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_0 - a_1 & a_1 - a_2 & a_2 - a_3 & a_3 - a_4 \\ -a_2 & a_0 - a_3 & a_1 - a_4 & a_2 \\ -a_3 & -a_4 & a_0 & a_1 \\ -a_4 & & & a_0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} a_0 - a_1 + a_2 - a_3 + a_4 & & & \\ -a_2 + a_3 - a_4 & a_0 - a_2 & a_1 - a_3 & a_2 - a_4 \\ a_4 - a_3 & -a_3 & a_0 - a_4 & a_1 \\ -a_4 & -a_4 & & a_0 \end{bmatrix}.$$

Általános esetben legyen a jobb oldalon lévő hasonlósági mátrix egy felső bidiagonális mátrix, amelynek a főátlójában 1-esek, a fölötte lévő átlóban pedig ∓ 1-esek vannak (ahol $\mp$ az ellenkező előjelt jelenti, mint a $D^{\pm}_{n,n}(f)$ indexe). Ennek az inverze (a bal szélső mátrix) egy felső háromszögmátrix, melynek elemei $t_{i,j} = (\pm 1)^{j-i}$, ha $i \leq j$. A kiindulási (középső) mátrix elemei a 3.2.1. Definíció alapján $d_{i,j} = a_{j-i} \pm a_{i+j-1}$. Ekkor könnyű látni, hogy a szorzat bal felső eleme, ahogy a példában is, valóban $f(\pm 1)$:

$$d'_{1,1} = \sum_{l=1}^{n} (\pm 1)^{l-1} d_{l,1} = a_0 + \sum_{l=1}^{n} (\pm 1)^{l} a_l = f(\pm 1).$$

Most kiszámítjuk az eredményt a második oszloptól kezdve ($j \geq 2$):

$$\begin{aligned} d'_{i,j} &= \sum_{l=i}^{n} (\pm 1)^{l-i} (d_{l,j} \mp d_{l,j-1}) = \\ &= \sum_{l=i}^{n} \big((\pm 1)^{l-i} (a_{j-l} - a_{l+j-2}) - (\pm 1)^{l-i+1} (a_{j-l-1} - a_{l+j-1}) \big) = \\ &= (\pm 1)^{i-i} (a_{j-i} - a_{i+j-2}) - (\pm 1)^{n-i+1} (a_{j-n-1} - a_{n+j-1}) = \\ &= a_{j-i} - a_{i+j-2}. \end{aligned}$$

Ha $i \geq 2$, akkor ezzel megmutattuk, hogy a jobb alsó $(n-1) \times (n-1)$-es aldetermináns valóban $D^{-}_{n,n-1}(f)$, ha pedig $i = 1$, akkor ezek az elemek 0-k. A determináns szabályai miatt ebből már következik a (3.2.6) felbontás.

3. A tétel harmadik részéhez a (3.2.1) egy módosított változatát kell bizonyítani, a már bizonyított (3.2.2) és (3.2.3) felhasználásával. A (3.2.1)-beli $\{1, \ldots, n\}$ és $\{1, \ldots, n-1\}$ indexhalmazokat lecseréljük valamilyen K és K' részhalmazukra:

$$\forall k \in K : D^{\pm}_{n,k}(f) > 0 \iff |a_n| < |a_0| \ \wedge \ \forall k \in K' \colon D^{\pm}_{n-1,k}(g) > 0. \tag{3.2.7}$$

Két bizonyítandó állítás van, a tétel 3/(a) és 3/(b) része. Ezt és az n paritását figyelembe véve négy eset van:

$$\begin{aligned} &\text{(a)},\ 2 \mid n: \quad K = \{2, 4, \ldots, n-2, n\},\ K' = \{1, 3, \ldots, n-3, n-1\}, \\ &\text{(a)},\ 2 \nmid n: \quad K = \{1, 3, \ldots, n-2, n\},\ K' = \{2, 4, \ldots, n-3, n-1\}, \\ &\text{(b)},\ 2 \mid n: \quad K = \{1, 3, \ldots, n-1, n\},\ K' = \{2, 4, \ldots, n-2, n-1\}, \\ &\text{(b)},\ 2 \nmid n: \quad K = \{2, 4, \ldots, n-1, n\},\ K' = \{1, 3, \ldots, n-2, n-1\}. \end{aligned}$$

A (b) résznél azért hagytuk benne a $k = n$-et, hogy lecserélhessük $f(\pm 1) > 0$-ra ugyanúgy, mint a tétel második részének bizonyításában.

A (3.2.7) „$\Longleftarrow$” iránya mind a négy esetben igaz, mivel (3.2.2) és (3.2.3) miatt elég hozzá az, hogy $\forall k \geq 2 : k \in K \Longleftrightarrow k-1 \in K'$. A „$\Longrightarrow$” irány viszont csak akkor működik, ha $1 \in K$, ami a négyből csak két esetben teljesül. A másik kettőben megmutatjuk, hogy a $k = 2$ esetből, azaz hogy $D^{\pm}_{n,2}(f) > 0$, következik a $k = 1$ eset, $D^{\pm}_{n,1}(f) > 0$:

$$\begin{aligned} D^{\pm}_{n,2}(f) = a_0(a_0 \pm a_{n-1}) - a_n(a_n \pm a_1) > 0 \iff \\ \iff a_0^2 - a_n^2 > |a_0 a_{n-1} - a_n a_1| \implies \\ \implies a_0^2 - a_n^2 > 0 \iff |a_n| < |a_0| \iff D^{\pm}_{n,1}(f) > 0, \end{aligned}$$

és ezzel befejeztük (3.2.7), és így a tétel harmadik részének bizonyítását. ■

Kiemeljük még egyszer, hogy a tétel csak valós együtthatós polinomokra igaz. Ezt kulcsfontosságú módon kihasználtuk a (3.2.2) ekvivalencia bizonyításában, ahol felbontottuk az $|a_n| < a_0$ feltételt $a_0 \pm a_n > 0$-ra.

3.2.2. Az eldöntési algoritmus

Az alábbiakban megvizsgáljuk az előző szakaszban bebizonyított 3.2.2. Tételt hatékonysági szempontból. Először szemléltetjük a működését egy példán keresztül.

3.2.3. Példa. Legyen a 3.1.8. Példából $f(x) := 2x^4 + x^3 - 14x^2 - 4x + 24$, ekkor a 3.2.2. Tételben szereplő mennyiségek a következők:

$$D_1^-(f) = 22, \qquad D_2^-(f) = \begin{vmatrix} 23 & -6 \\ -2 & 24 \end{vmatrix} = 540, \qquad D_3^-(f) = \begin{vmatrix} 38 & -5 & -16 \\ -1 & 22 & -4 \\ -2 & 0 & 24 \end{vmatrix} = 19200,$$

$$D_1^+(f) = 26, \qquad D_2^+(f) = \begin{vmatrix} 25 & -2 \\ 2 & 24 \end{vmatrix} = 604, \qquad D_3^+(f) = \begin{vmatrix} 10 & -3 & -12 \\ 1 & 26 & -4 \\ 2 & 0 & 24 \end{vmatrix} = 6960,$$

$$f(-1) = 15, \qquad f(+1) = 9.$$

Mivel mindegyik mennyiség pozitív, ezért a polinom expanzív. Az is látható, hogy ezek az értékek jóval kisebbek, mint amiket a Schur–Cohn-teszt produkál (3.1.8. Példa).

Ha a 3.2.2. Tételt a gyakorlatban szeretnénk alkalmazni, akkor célszerű először a legegyszerűbb feltételeket kiértékelni, majd rendre az egyre bonyolultabbakat, hogy ha valamelyik nem teljesül, akkor minél hamarabb leállhassunk. A tétel 1. és 2. része alapján a következő algoritmust írhatjuk fel:

3.2.4. Algoritmus: Expanzivitás eldöntése a 3.2.2. Tétel alapján

Bemenet: $f \in \mathbb{Z}[x]$ polinom
Kimenet: expanzív-e
if $a_0 = 0$ **then**
  return false
if $a_0 < 0$ **then**
  $f := -f$
if $|a_n| \geq a_0$ **then**
  return false
if $f(1) \leq 0 \lor f(-1) \leq 0$ **then**
  return false
for $k := 2$ **to** $n-1$ **do**
  if $D_k^-(f) \leq 0 \lor D_k^+(f) \leq 0$ **then**
    return false
return true

Az algoritmus legelején biztosítjuk a tétel $a_0 > 0$ feltételét. A $k = 1$-es D-feltételek az ekvivalens $|a_n| < a_0$ formában szerepelnek, és ezzel kezdjük, mert ezt a leggyorsabb kiértékelni. Gyorsíthatjuk az eljárást a tétel 3. részének segítségével, miszerint a D-feltételeket elég csak minden második k-ra ellenőrizni ($k = \ldots, n-5, n-3, n-1$-re). Az első néhány k-t azonban ennek ellenére érdemes paritástól függetlenül megvizsgálni, mert gyorsan adhatnak negatív eredményt, és nem kerülnek sokba.

Most megvizsgáljuk az algoritmus futási idejét a legrosszabb esetben, azaz ha minden feltételt ellenőrizni kell, például mert a polinom expanzív. A legfontosabb lépés, amit elemezni kell, a $D_k^{\pm}(f)$-ek kiszámítása. A determinánsok mérete legfeljebb $(n-1)\times(n-1)$, elemei pedig $a_i \pm a_j$ alakúak (lásd a 3.2.1. Definíciót), vagyis abszolút értékük legfeljebb $2H$, ahol $H := H(f)$ az f magassága (3.1.2. Definíció). Ezért a determinánsok általános tulajdonságaiból (lásd: (2.2.16)) könnyen látható, hogy $\log \left|D_k^{\pm}(f)\right| \leq n \log(2nH)$, vagyis méretük valóban polinomiális. Ez azonban még nem elegendő, mert vigyázni kell arra is, hogy a determinánsok kiszámítása során se legyen együtthatórobbanás, ami nem megfelelő algoritmus esetén előfordulhat (pl. a Gauss-elimináció naiv alkalmazásakor). Ezt elkerülendő, a 2.3. szakaszban tárgyalt Bareiss-algoritmust használjuk. Ennek futási ideje $O(n^3 \operatorname{Mul}(nA + n\log n)))$ (2.3.3), ahol A az elemek hosszának felső korlátja, jelen esetben $A = \log(2H)$, a $\operatorname{Mul}(\cdot)$ jelölést pedig a 2.2.2. szakasz elején definiáltuk az egész számok szorzásának bonyolultságára. A teljes eldöntési algoritmus $O(n)$ ilyen

determinánst számít ki, ezért futási ideje összességében:

$$\mathrm{T}_{n,H}(D\text{-exp}) = O\big(n^4\,\mathrm{Mul}(n\log(nH))\big),$$

ami hagyományos egészszorzás ($\mathrm{Mul}(X) = X^2$) esetén $O\big(n^6\log^2(nH)\big)$-t jelent. Megmutattuk tehát, hogy a 3.2.2. Tétel polinomiális időben alkalmazható.

3.2.3. Általánosítás: D-polinomok

Ebben a szakaszban általánosítjuk a $D_k^{\pm}(f)$ determinánsokat polinomokká. Beírunk a definiciójukba egy új változót, és megvizsgáljuk az így kapott determináns-polinomok tulajdonságait. Például segítségükkel feltételeket tudunk adni az expanzivitás általánosítására, azaz hogy minden gyök abszolút értéke nagyobb egy adott konstansnál (ami expanzivitásnál 1).

3.2.5. Definíció. Legyen $\widetilde{D}_k^{\pm}(f)(x)$ hasonló, mint $D_k^{\pm}(f)$ a 3.2.1. Definíció szerint, de minden a_j helyére a_jx^j-t írunk, azaz a determináns a következő elemekből áll:

$$\widetilde{d}_{ij} = a_{j-i}x^{j-i} \pm a_{i+j+n-k-1}x^{i+j+n-k-1} \qquad (1 \le i, j \le k).$$

Az így kapott kifejezéseket *D-polinom*oknak nevezzük.

Például $n = 7$-re és $k = 6$-ra:

$$\widetilde{D}_6^{-}(f)(x) = \begin{vmatrix} a_0 - a_2x^2 & a_1x - a_3x^3 & a_2x^2 - a_4x^4 & a_3x^3 - a_5x^5 & a_4x^4 - a_6x^6 & a_5x^5 - a_7x^7 \\ -a_3x^3 & a_0 - a_4x^4 & a_1x - a_5x^5 & a_2x^2 - a_6x^6 & a_3x^3 - a_7x^7 & a_4x^4 \\ -a_4x^4 & -a_5x^5 & a_0 - a_6x^6 & a_1x - a_7x^7 & a_2x^2 & a_3x^3 \\ -a_5x^5 & -a_6x^6 & -a_7x^7 & a_0 & a_1x & a_2x^2 \\ -a_6x^6 & -a_7x^7 & & & a_0 & a_1x \\ -a_7x^7 & & & & & a_0 \end{vmatrix}.$$

A sima D-k és a D-polinomok között közvetlen kapcsolat van: egyrészt természetesen $\widetilde{D}_k^{\pm}(f)(1) = D_k^{\pm}(f)$, másrészt pedig ha $f_y(x) := f(xy)$, akkor $D_k^{\pm}(f_y) = \widetilde{D}_k^{\pm}(f)(y)$.

Ezt felhasználva a következőt kapjuk:

3.2.6. Következmény. *Ha f egy valós együtthatós polinom, melyre $a_0 > 0$, akkor bármely $s > 0$ konstansra f-nek pontosan akkor lesz minden α_i gyökére $|\alpha_i| > s$, ha a 3.2.2. Tételhez hasonló feltételek teljesülnek, de minden $D_k^{\pm}(f)$ helyett $\widetilde{D}_k^{\pm}(f)(s)$-et írunk, és $f(\pm 1)$ helyett $f(\pm s)$-et.*

Bizonyítás. Alkalmazzuk a 3.2.2. Tételt $f_s(x) := f(sx)$-re, ekkor az $|\alpha_i| > s$ feltételek f-re ekvivalensek az f_s expanzivitásával. ■

A D-polinomoknak további érdekes tulajdonságaik vannak:

3.2.7. Lemma.

1. *A $\widetilde{D}_k^{\pm}(f)(x)$ konstans tagja a_0^k, főtagja pedig $a_n^k x^{kn}$, de az utóbbinak van egy előjele, ami a paraméterektől (k és $\pm$) függ.*
2. *A $k = n-1, n-3, n-5, \ldots$ paraméterű $\widetilde{D}_k^{\pm}(f)(x)$-ek kifejtésében kizárólag páros kitevőjű x-ek szerepelnek.*
3. *A $\widetilde{D}_{n-1}^{-}(f)(x^{1/2})$ polinomnak a következő $\binom{n}{2}$ gyöke van: $\alpha_i\alpha_j$ minden $i < j$-re.*

Bizonyítás. (1) A $\widetilde{D}_k^{\pm}(f)(x)$ determinánsnak mind a k sorában egyetlen legkisebb és egyetlen legnagyobb x-hatvány van, a_0 és $\pm a_n x^n$. Az előbbiek a főátlóban vannak, így az egész determináns konstans tagját ők adják: a_0^k, az utóbbiak pedig a mellékátlóban helyezkednek el, így a főtag $(-1)^{\lfloor k/2 \rfloor}(\pm 1)^k a_n^k x^{kn}$ lesz.

(2) Szorozzuk meg a $\widetilde{D}_k^{\pm}(f)(x)$ determináns minden i. sorát x^{i-1}-nel, és osszuk le minden j. oszlopát x^{j-1}-nel. Ezáltal a determináns nem változik, az új elemek pedig a következők lesznek (vö. 3.2.5. Definíció):

$$\widetilde{d}'_{ij} = a_{j-i} \pm a_{i+j+n-k-1} x^{2i+n-k-1},$$

amelyből már nyilvánvaló, hogy ha $n-k-1$ páros, akkor x-nek csak páros kitevői szerepelnek benne. (Megjegyzés: ha $x = 0$, akkor ez a transzformáció nem működik, de akkor a két determináns triviálisan egyenlő.)

(3) Először alkotunk egy n^2-fokú polinomot, amelynek gyökei $\alpha_i\alpha_j$ $(1 \le i, j \le n)$. A 2.1.8. Tételből és a 2.1.6. Definícióból következik, hogy a következő polinom ilyen:

$$F(x) = \operatorname{res}_y\left(y^n f\left(\frac{x}{y}\right), f(y)\right) = (-1)^n a_n^{2n} \prod_{i=1}^{n}\prod_{j=1}^{n}(x - \alpha_i\alpha_j).$$

A rezultánsok kifejezhetők a Sylvester-mátrix determinánsával [14, Lemma 3.3.4.], ami jelen esetben $2n \times 2n$-es, és a következőképpen néz ki:

$$F(x) = \begin{vmatrix}
a_0 & & & & a_n & & & \\
a_1 x & a_0 & & & a_{n-1} & a_n & & \\
a_2 x^2 & a_1 x & \ddots & & \vdots & a_{n-1} & \ddots & \\
\vdots & a_2 x^2 & \ddots & a_0 & a_1 & \vdots & \ddots & a_n \\
a_n x^n & \vdots & \ddots & a_1 x & a_0 & a_1 & & a_{n-1} \\
 & a_n x^n & & a_2 x^2 & & a_0 & \ddots & \vdots \\
 & & \ddots & \vdots & & & \ddots & a_1 \\
 & & & a_n x^n & & & & a_0
\end{vmatrix}.$$

A törtkitevők elkerülése végett $F(x)$ helyett $F(x^2)$-tel számolunk. Megmutatjuk, hogy ezt fel tudjuk bontani a következőképpen:

$$F(x^2) = \widetilde{D}_n^+(f)(x) \cdot \widetilde{D}_n^-(f)(x). \tag{3.2.8}$$

A fenti mátrixot egy hasonlósági transzformációval átalakítjuk úgy, hogy a determináns ne változzon, és az eredményből már nyilvánvaló legyen a felbontás. Ezt először bemutatjuk egy példán ($n = 3$):

$$\begin{bmatrix} x^2 & & & & & \\ & x & & & & \\ & & 1 & & & \\ & & x & 1 & & \\ & x^2 & & & x^{-1} & \\ x^3 & & & & & x^{-2} \end{bmatrix} \cdot \begin{bmatrix} a_0 & & & a_3 & & \\ a_1x^2 & a_0 & & a_2 & a_3 & \\ a_2x^4 & a_1x^2 & a_0 & a_1 & a_2 & a_3 \\ a_3x^6 & a_2x^4 & a_1x^2 & a_0 & a_1 & a_2 \\ & a_3x^6 & a_2x^4 & & a_0 & a_1 \\ & & a_3x^6 & & & a_0 \end{bmatrix} \cdot \begin{bmatrix} x^{-2} & & & & & \\ & x^{-1} & & & & \\ & & 1 & & & \\ & & -x & 1 & & \\ & -x^2 & & & x & \\ -x^3 & & & & & x^2 \end{bmatrix} =$$

$$= \begin{bmatrix} a_0 & & -a_3x^3 & a_3x^2 & & \\ a_1x & a_0 - a_3x^3 & -a_2x^2 & a_2x & a_3x^2 & \\ a_2x^2 - a_3x^3 & a_1x - a_2x^2 & a_0 - a_1x & a_1 & a_2x & a_3x^2 \\ & & & a_1x + a_0 & a_2x^2 + a_1x & a_3x^3 + a_2x^2 \\ & & & a_2x^2 & a_3x^3 + a_0 & a_1x \\ & & & a_3x^3 & & a_0 \end{bmatrix}.$$

Általánosan jelöljük M-mel az eredeti mátrixot, és alkalmazzunk rá az $UMV = N$ ($U = V^{-1}$) hasonlósági transzformációt a következőkkel:

$$m_{ij} = \begin{cases} a_{i-j}x^{2i-2j} & (1 \le j \le n), \\ a_{j-i} & (n+1 \le j \le 2n), \end{cases}$$

$$u_{ij}; v_{ij} = \begin{cases} x^{n-j}; x^{j-n} & (1 \le i = j \le n), \\ x^{n-j+1}; x^{j-n-1} & (n+1 \le i = j \le 2n), \\ x^{n-j+1}; -x^{n-j+1} & (i = 2n-j+1, 1 \le j \le n), \\ 0 & (\text{egyébként}). \end{cases}$$

A mátrixszorzások elvégzése után a következőket kapjuk:

$$(UM)_{ij} = \begin{cases} a_{i-j}x^{n+i-2j} & (1 \le i, j \le n), \\ a_{j-i}x^{n-i} & (1 \le i \le n, n+1 \le j \le 2n), \\ a_{i-j}x^{n+i-2j+1} + a_{2n-i-j+1}x^{3n-i-2j+2} & (n+1 \le i \le 2n, 1 \le j \le n), \\ a_{j-i}x^{n-i+1} + a_{i+j-2n-1}x^{i-n} & (n+1 \le i, j \le 2n), \end{cases}$$

$$n_{ij} = (UMV)_{ij} = \begin{cases} a_{i-j}x^{i-j} - a_{2n-i-j+1}x^{2n-i-j+1} & (1 \le i, j \le n), \\ a_{j-i}x^{j-i-1} & (1 \le i \le n, n+1 \le j \le 2n), \\ 0 & (n+1 \le i \le 2n, 1 \le j \le n), \\ a_{j-i}x^{j-i} + a_{i+j-2n-1}x^{i+j-2n-1} & (n+1 \le i, j \le 2n). \end{cases}$$

Látható általában is, amit a példán is láttunk, hogy a jobb alsó negyed pontosan $\widetilde{D}_n^+(f)(x)$, a bal felső negyed $\widetilde{D}_n^-(f)(x)$, csak fejjel lefelé (azaz középpontosan tükrözve), a bal alsó negyed pedig nulla. (Ismét meg kell jegyeznünk, hogy $x = 0$ esetén a transzformáció nem működik, de ekkor triviálisan $\det M = \det N$.) Ezzel megmutattuk a (3.2.8) felbontást.

A 3.2.2. Tétel második részének bizonyításából tudjuk, hogy $D_n^{\pm}(f) = f(\pm 1)\, D_{n-1}^-(f)$. Ezt ki tudjuk terjeszteni $\widetilde{D}$-ra, ismerve a D és a $\widetilde{D}$ közötti kapcsolatot (lásd a 3.2.5. Definíció utáni megjegyzést): $\widetilde{D}_n^{\pm}(f)(x) = f(\pm x)\, \widetilde{D}_{n-1}^-(f)(x)$. Ennek segítségével tovább tudjuk finomítani a (3.2.8) felbontást:

$$F(x^2) = f(x)f(-x)\Big(\widetilde{D}_{n-1}^-(f)(x)\Big)^2. \tag{3.2.9}$$

Vizsgáljuk meg ennek az első felét:

$$f(x)f(-x) = a_n \prod_{i=1}^{n}(x - \alpha_i) \cdot a_n \prod_{i=1}^{n}(-x - \alpha_i) = (-1)^n a_n^2 \prod_{i=1}^{n}(x^2 - \alpha_i^2).$$

Térjünk vissza $F(x^2)$-ről $F(x)$-re, azaz x helyére írjunk mindenütt $x^{1/2}$-t. Látható, hogy $f(x^{1/2})f(-x^{1/2})$ egyrészt polinom, másrészt a gyökei α_i^2 $(1 \le i \le n)$, azaz pontosan az $f(x)$ gyökeinek négyzetei. Ezért $F(x)$ összes többi gyöke kétszeres ($\alpha_i\alpha_j = \alpha_j\alpha_i$ minden $i \ne j$-re), tehát a (3.2.9) felbontás alapján $\widetilde{D}_{n-1}^-(f)(x^{1/2})$ gyökei pontosan ezek fele lesz, vagyis $\alpha_i\alpha_j$ $(i < j)$. ■

3.3. Alsó korlátok az expanzivitási hézagra

Ebben a szakaszban arra a kérdésre keressük a választ, hogy egy egész együtthatós expanzív polinomnak milyen kicsik lehetnek a gyökei, azaz mennyire kerülhetnek közel a komplex egységkörhöz. Ennek mérésére a következő mennyiséget használjuk:

3.3.1. Definíció. Ha f egy expanzív polinom (azaz minden α_i gyökére $|\alpha_i| > 1$), akkor az alábbi számot f *expanzivitási hézag*jának nevezzük:

$$\varepsilon := \min_{i=1}^{n} |\alpha_i| - 1.$$

A kérdés tehát az, hogy ε mennyire lehet kicsi. Ez a probléma hasonlít a híres Lehmer-sejtésre [24], amely szerint egész együtthatós polinomok Mahler-mértéke (3.1.2. Definíció) vagy 1, vagy legalább $\mu \approx 1{,}176$, más szóval ha a polinomnak vannak 1-nél nagyobb abszolút értékű gyökei, akkor ezek szorzata nem lehet tetszőlegesen közel az egységkörhöz.

Az általunk megfogalmazott kérdésnél azonban nem ez a helyzet, ahogy azt a következő két egyszerű példa is mutatja:

$$f(x) = x^n - 2 \qquad (n \in \mathbb{N}^+) \qquad \Longrightarrow \qquad \varepsilon = \sqrt[n]{2} - 1 \xrightarrow{n\to\infty} 0,$$

$$f(x) = Ax - (A+1) \qquad (A \in \mathbb{N}^+) \qquad \Longrightarrow \qquad \varepsilon = \frac{1}{A} \xrightarrow{A\to\infty} 0.$$

Látható, hogy az expanzivitási hézag mindkét sorozatban tetszőlegesen kicsi lehet. Az elsőnél a korlátlan fokszám, a másodiknál pedig a korlátlan méretű együtthatók tették lehetővé ezt. De mi a helyzet akkor, ha mind a fokszámot, mind az együtthatókat korlátozzuk? Egész együtthatókról lévén szó, csak véges sok ilyen polinom van, tehát kell hogy legyen ε-nak pozitív alsó korlátja. Ebben a szakaszban ezzel fogunk foglalkozni: keresünk egy explicit alsó korlátot a fokszám és az együtthatók méretének függvényében. Az utóbbi számszerűsítésére több lehetőségünk is van, például használhatjuk a polinom magasságát ($H(f)$) vagy hosszúságát ($L(f)$), a 3.1.2. Definíció alapján.

Az expanzivitási hézagot visszavezetjük valós algebrai számok távolságára. Ha ugyanis az egységkörhöz legközelebbi gyök α, akkor $\varepsilon = |\alpha| - 1$, ami két valós algebrai szám ($|\alpha|$ és 1) távolsága. A probléma az, hogy hogyan tudjuk kezelni $|\alpha|$-t, például felírni a minimálpolinomját. Valós α esetén ez könnyű, mert $|\alpha| = \pm\alpha$, de tetszőleges komplex α-ra nehéz. Ehelyett azt használjuk, hogy $|\alpha| = \sqrt{\alpha\overline{\alpha}}$, így ha $|\alpha|$ közel van az 1-hez, akkor $|\alpha| - 1 \approx \frac{\alpha\overline{\alpha}-1}{2}$, vagyis $|\alpha|$ helyett elég az $\alpha\overline{\alpha}$ algebrai számmal foglalkozni. Ez utóbbinak pedig, a 3.2.7. Lemma alapján, az egyik D-polinomból képezhető a minimálpolinomja.

3.3.1. Liouville-típusú korlátok

Algebrai számok közötti távolság alsó becslésére gyakran használt eszköz a Liouville-egyenlőtlenség [34, Prop. 3.14]:

3.3.2. Tétel. *Ha f és g egész együtthatós polinom, $f(\alpha) = 0$ és $g(\alpha) \neq 0$, akkor*

$$\frac{1}{|g(\alpha)|} \leq L(g)^{n-1} M(f)^{\deg g},$$

ahol $L(g)$ a g hosszúsága, és $M(f)$ az f Mahler-mértéke (3.1.2. Definíció).

A Liouville-egyenlőtlenség segítségével becsülni tudjuk az expanzivitási hézagot. A formulák egyszerűsítése miatt alsó korlát helyett a reciprokra adunk felső korlátot.

3.3.3. Tétel. *Ha f egy egész együtthatós expanzív polinom és α egy gyöke, akkor*

$$\frac{1}{|\alpha| - 1} \leq \begin{cases} 2^{n-1}|a_0| & (\alpha \in \mathbb{R}), \\ 2^{\binom{n}{2}}|a_0|^{n-1} + 1 & (\alpha \in \mathbb{C} \setminus \mathbb{R}). \end{cases}$$

Bizonyítás. Ha α valós, akkor alkalmazzuk a Liouville-egyenlőtlenséget (3.3.2. Tétel) $g(x) = x \pm 1$-re, kihasználva, hogy expanzív polinomra $M(f) = |a_0|$:

$$\frac{1}{|\alpha \pm 1|} \leq 2^{n-1}|a_0|, \tag{3.3.1}$$

és ebből megkapjuk az állítás valós részét.

Ha α nem valós, akkor (3.3.1)-et alkalmazzuk α helyett $\alpha\overline{\alpha}$-ra. Mivel α és $\overline{\alpha}$ is gyöke f-nek, ezért $\alpha\overline{\alpha}$ a 3.2.7. Lemma alapján gyöke $F(x) := \widetilde{D}^{-}_{n-1}(f)(x^{1/2})$-nek. Ez utóbbi polinom $\binom{n}{2}$-edfokú, expanzív, és a konstans tagja a_0^{n-1}, ezért (3.3.1)-ből a „$-$” előjel használatával a következő egyenlőtlenség lesz:

$$\frac{1}{|\alpha\overline{\alpha} - 1|} \leq 2^{\binom{n}{2}-1}|a_0|^{n-1}.$$

Ezt átrendezve azt kapjuk, hogy $|\alpha|^2 = \alpha\overline{\alpha} \geq 1 + \frac{1}{B}$, ahol B a fenti formula jobb oldala. A bizonyítást a következő elemi egyenlőtlenség alkalmazásával fejezzük be:

$$\sqrt{1 + \frac{1}{B}} > 1 + \frac{1}{2B + 1}, \tag{3.3.2}$$

ahol B tetszőleges pozitív szám. ■

Erősebb korlátot adhatunk, ha a konstans tag mellett a főegyütthatót is felhasználjuk:

3.3.4. Tétel. *Ha f egy egész együtthatós expanzív polinom és α egy gyöke, akkor*

$$\frac{1}{|\alpha| - 1} \leq \begin{cases} 2^{n-2}(|a_0| + |a_n|) & (\alpha \in \mathbb{R}), \\ 2^{\binom{n-1}{2}}(|a_0| + |a_n|)^{n-1} + 1 & (\alpha \in \mathbb{C} \setminus \mathbb{R}). \end{cases}$$

Bizonyítás. Az előző tételben felhasználtuk a Liouville-egyenlőtlenség egy speciális esetét, (3.3.1)-et. Ezt le fogjuk cserélni egy erősebb állításra, de ehhez először bebizonyítjuk.

Osszuk le az $f(x) = a_n(x-\alpha_1)(x-\alpha_2)\dots(x-\alpha_n)$ alakot az első gyöktényezőjével, helyettesítsünk be $x = \pm 1$-et, és vegyük az egész abszolút értékét:

$$\frac{|f(\pm 1)|}{|\pm 1 - \alpha_1|} = |a_n| \prod_{i=2}^{n} |\pm 1 - \alpha_i|.$$

A bal oldalt alulról becsülhetjük azzal, hogy $|f(\pm 1)| \geq 1$, mert f expanzív (így $f(\pm 1) \neq 0$) és egész együtthatós. A jobb oldalt felülről becsüljük:

$$\frac{1}{|\pm 1 - \alpha_1|} \leq \frac{|f(\pm 1)|}{|\pm 1 - \alpha_1|} \leq |a_n| \prod_{i=2}^{n} (1 + |\alpha_i|) \leq$$
$$\leq \frac{1}{2}|a_n| \prod_{i=1}^{n} (1 + |\alpha_i|) \leq \frac{1}{2}|a_n| \prod_{i=1}^{n} 2|\alpha_i| = 2^{n-1}|a_0|,$$

ebből pedig már következik (3.3.1).

A (3.3.1) javításához lecseréljük a fenti levezetésben az utolsó egyenlőtlenséget. Az előtte lévő kifejezést elnevezzük $\frac{1}{2}\hat{f}(1)$-nek, ehhez definiálunk egy $\hat{f}$ polinomot úgy, hogy a gyökei $-|\alpha_i|$ legyenek, pontosabban:

$$\hat{f}(x) = \hat{a}_n x^n + \ldots + \hat{a}_1 x + \hat{a}_0 := |a_n|(x + |\alpha_1|)(x + |\alpha_2|)\dots(x + |\alpha_n|).$$

Az előző levezetésből tehát, ha az utolsó egyenlőtlenséget már nem alkalmazzuk, az következik, hogy $\frac{1}{|\alpha \pm 1|} \leq \frac{1}{2}\hat{f}(1)$, és innen fogjuk másképp folytatni.

Ehhez először bebizonyítunk egy állítást, amelyet többször fogunk alkalmazni: ha f expanzív polinom (tetszőleges komplex együtthatókkal), akkor

$$|f(x)| \leq (1 + |x|)^{n-1}(|a_n||x| + |a_0|). \tag{3.3.3}$$

Ez könnyen kijön a 3.1.4. Lemma alkalmazásával:

$$|f(x)| \leq \sum_{k=0}^{n} |a_k||x|^k \leq \sum_{k=0}^{n} \left(\binom{n-1}{k-1} |a_n| + \binom{n-1}{k} |a_0| \right) |x|^k =$$
$$= \sum_{k=0}^{n-1} \binom{n-1}{k} |x|^k (|a_n||x| + |a_0|),$$

és a binomiális tétel kiadja (3.3.3)-at.

Ezután már csak alkalmaznunk kell (3.3.3)-at az expanzív $\hat{f}$-ra, és az $\frac{1}{|\alpha \pm 1|} \leq \frac{1}{2}\hat{f}(1)$ egyenlőtlenség folytatásával megkapjuk a tétel állítását valós α-ra.

Ha α nem valós, akkor hasonlóan kezdjük el, mint fent, csak α és f helyett $\alpha\overline{\alpha}$-ra és F-re, mint az előző tétel bizonyításában. Legyen $\hat{F}$ hasonló, mint $\hat{f}$, csak f helyett

F-ből. Ekkor a fentihez hasonlóan azt kapjuk, hogy $\frac{1}{\alpha\overline{\alpha}-1} \leq \frac{1}{2}\hat{F}(1)$, és ezt kell tovább becsülni. $\hat{F}(1)$-et a következőképpen írhatjuk fel:

$$\hat{F}(1) = \prod_{m=2}^{n}\left(|a_n|\prod_{i=1}^{m-1}(1+|\alpha_i||\alpha_m|)\right) = \prod_{m=2}^{n}\left(|\alpha_m|^{m-1}\hat{f}_{m-1}\left(\frac{1}{|\alpha_m|}\right)\right),$$

ahol $\hat{f}_m(x) := |a_n|(x+|\alpha_1|)\dots(x+|\alpha_m|)$. Ez utóbbi is egy expanzív polinom, vagyis alkalmazhatjuk rá (3.3.3)-at, folytatva az előzőt:

$$\leq \prod_{m=2}^{n}\big((1+|\alpha_m|)^{m-2}|a_n|(1+|\alpha_1|\dots|\alpha_m|)\big) =$$
$$= \prod_{l=2}^{n}|a_n|(1+|\alpha_1|\dots|\alpha_l|)(1+|\alpha_{l+1}|)\dots(1+|\alpha_n|).$$

Ha a $\prod$-ban lévő szorzatot egy expanzív polinom gyöktényezős alakjának tekintjük, amelybe behelyettesítettünk $x=1$-et, akkor alkalmazhatjuk rá (3.3.3)-at ismét:

$$\leq \prod_{l=2}^{n}2^{n-l}|a_n|(1+|\alpha_1|\dots|\alpha_n|) = 2^{\binom{n-1}{2}}(|a_n|+|a_0|)^{n-1}.$$

Ezután a bizonyítást hasonlóan fejezzük be, mint a 3.3.3. Tételét. ■

3.3.2. A determinásokból adódó korlátok

Az expanzivitási hézag eddig bemutatott becslései (3.3.3. és 3.3.4. Tétel) a Liouville-egyenlőtlenségen és az expanzív polinomok alaptulajdonságain alapultak. Az együtthatók közül csak a konstans tag és a főegyüttható szerepelt bennük, viszont az (n-től függő) szorzó hatalmas volt (pl. $2^{\binom{n}{2}}$). Ebben a szakaszban olyan becsléseket adunk, amelyekben ez a szorzó kisebb, viszont cserébe megjelenik az összes többi együttható is. Két tételt bizonyítunk, az egyikben a polinom magassága ($H(f)$), a másikban a hosszúsága ($L(f)$) szerepel (3.1.2. Definíció). A tételekben használjuk a D-feltételek és a D-polinomok 3.2. szakaszban tárgyalt tulajdonságait.

3.3.5. Tétel. *Ha f egy egész együtthatós expanzív polinom és α egy gyöke, akkor*

$$\frac{1}{|\alpha|-1} \leq \begin{cases} \binom{n+1}{2}H(f)+\frac{n}{2} & (\alpha\in\mathbb{R}),\\ \binom{n}{2}n!H(f)^{n-1}+\binom{n}{2}+1 & (\alpha\in\mathbb{C}\setminus\mathbb{R}).\end{cases}$$

Bizonyítás. Mivel $|f(\pm1)|\geq 1$, ezért ha egy ε pozitív valós számra megmutatjuk, hogy $|f(\pm(1+\varepsilon))-f(\pm1)|<1$, akkor f-nek nincs $1+\varepsilon$ abszolút értékű valós gyöke. Ha ez minden $\varepsilon<\varepsilon_0$-ra igaz, akkor minden valós gyök abszolút értéke legalább $1+\varepsilon_0$.

$f(1+\varepsilon)$-t a következőképpen lehet (véges) hatványsorba fejteni:

$$f(1+\varepsilon) = f(1) + f'(1)\varepsilon + \frac{f''(1)}{2}\varepsilon^2 + \ldots + \frac{f^{(n)}(1)}{n!}\varepsilon^n.$$

$f(-1-\varepsilon)$-t hasonlóan, csak néhány előjel negatív lesz. Becsüljük meg az együtthatókat:

$$\left|\frac{f^{(k)}(\pm 1)}{k!}\right| = \left|\sum_{j=k}^{n}(\pm 1)^{j-k}\binom{j}{k}a_j\right| \leq H(f)\sum_{j=k}^{n}\binom{j}{k} = \binom{n+1}{k+1}H(f),$$

ezt felhasználva:

$$|f(\pm(1+\varepsilon)) - f(\pm 1)| = \left|\sum_{k=1}^{n}\frac{f^{(k)}(\pm 1)}{k!}\varepsilon^k\right| \leq \sum_{k=1}^{n}\binom{n+1}{k+1}H(f)\varepsilon^k < \sum_{k=1}^{\infty}(n+1)\left(\frac{n}{2}\varepsilon\right)^k H(f).$$

Könnyű megmutatni, hogy tetszőleges $A, B > 0$-ra:

$$\sum_{k=1}^{\infty}(A\varepsilon)^k B \leq 1 \quad \Longleftrightarrow \quad \varepsilon \leq \frac{1}{A(B+1)}, \tag{3.3.4}$$

és ezt alkalmazva megkapjuk az állítást valós α-ra.

Ha α nem valós, akkor akárcsak az előző tételekben, F-fel számolunk f helyett (lásd a 3.3.3. Tétel bizonyítását), és legyen $F(x) = A_N x^N + \ldots + A_2 x^2 + A_1 x + A_0$. Ennek fokszáma $N := \binom{n}{2}$, és $F(1) = D^{-}_{n-1}(f)$, ami felírható egy $(n-1)\times(n-1)$-es determinánsként, amelynek minden eleme vagy 0, vagy $\pm a_i$, vagy $a_i - a_j$ alakú (lásd a 3.2.1. Definíciót). Fejtsük ki ezt a determinánst teljesen (lásd (2.2.16)-ot), és szorozzuk ki az összes $a_i - a_j$ elemet is, így egy következő alakú kifejezést kapunk:

$$F(1) = \sum_{l=0}^{N} A_l = \sum_{i=1}^{T_n}\left(\pm\prod_{j=1}^{n-1} a_{s_{i,j}}\right), \tag{3.3.5}$$

ahol $s_{i,j}$ az i. tag azon tényezőjének indexe, amely a determináns j. oszlopából van, és T_n a kifejtésben szereplő tagok száma. Jelöljük továbbá $T_n^{(l)}$-lel az A_l-hez tartozó tagok számát, más szóval azon tagok számát, amelyekben az indexek összege l. Ekkor a fentiből $|A_l| \leq T_n^{(l)} H(f)^{n-1}$, összesen pedig:

$$|F(1)| \leq \sum_{l=0}^{N}|A_l| \leq \sum_{l=0}^{N} T_n^{(l)} H(f)^{n-1} = T_n H(f)^{n-1}. \tag{3.3.6}$$

T_n-re a következőképpen adhatunk korlátot. Illusztrációként tekinthetjük a 3.2.1. Definíció utáni példát. Fejtsük ki a determinánst soronként, az utolsó sortól az elsőig. Az utolsó sorban két tag van, így a kifejtéskor két választási lehetőség van. Az utolsó előtti sorban 4 tag van, de az egyik alatti tagot már kiválasztottuk, így kifejtéskor csak 3 választási lehetőség marad. Minden további sor két új tagot hoz be (az $a_i - a_j$ alakú elemeket duplán számítva), de mindig eggyel több oszlopot zárunk ki (ami egy vagy két tagot jelent), ezért mindig legfeljebb eggyel több választási lehetőség van, mint az alatta

lévő sorban. Ha ezt végigvisszük, akkor összesen legfeljebb $2 \cdot 3 \cdot 4 \cdot \ldots \cdot n$ lehetőségünk van tagokat alkotni a kifejtésben, tehát $T_n \leq n!$.

Következésképp (3.3.6)-ból azt kapjuk, hogy $\sum_{l=0}^{N} |A_l| \leq n!H(F)^{n-1}$. A tétel első részéhez hasonlóan folytatjuk, de f helyett F-re:

$$\frac{|F^{(k)}(1)|}{k!} = \left| \sum_{l=k}^{N} \binom{l}{k} A_l \right| \leq \binom{N}{k} \sum_{l=0}^{N} |A_l| \leq \binom{N}{k} n!H(f)^{n-1}.$$

Ehelyett azonban egy picit jobb felső korlátot adunk, egészen pontosan egy kettes szorzóval. Ehhez felhasználjuk a determináns (3.2.1. Definíció) szimmetriáját: ha az $a_0, a_1, \ldots, a_n$ sorozatot megfordítjuk, azaz minden a_i-t lecserélünk a_{n-i}-re, akkor a determinánsnak legfeljebb az előjele változhat. Ebből következik, hogy a kifejtésben minden $\prod a_i$ tagnak van egy párja, $\prod a_{n-i}$ (ami lehet önmaga is), esetleg eltérő előjellel. A D-polinomok definícióját (3.2.5. Definíció) felhasználva ebből arra jutunk, hogy A_l és A_{N-l} ugyanannyi $\prod a_i$ tagot tartalmaz, azaz $T_n^{(l)} = T_n^{(N-l)}$, tehát párosíthatjuk őket, és így a fentinél jobb korlátot tudunk adni:

$$\frac{|F^{(k)}(1)|}{k!} = \left| \sum_{l=0}^{N} \frac{1}{2} \left(\binom{l}{k} A_l + \binom{N-l}{k} A_{N-l} \right) \right| \leq$$

$$\leq \frac{1}{2} \max_{l=0}^{N} \left(\binom{l}{k} + \binom{N-l}{k} \right) \sum_{l=0}^{N} T_n^{(l)} H(f)^{n-1} \leq \frac{1}{2} \binom{N}{k} n!H(f)^{n-1}.$$

Ezt a korlátot alkalmazva:

$$|F(1+\varepsilon) - F(1)| = \left| \sum_{k=1}^{N} \frac{F^{(k)}(1)}{k!} \varepsilon^k \right| \leq \sum_{k=1}^{N} \frac{1}{2} \binom{N}{k} n!H(f)^{n-1} \varepsilon^k < \sum_{k=1}^{\infty} \left(\frac{N}{2} \varepsilon \right)^k n!H(f)^{n-1}.$$

A hatványsorra vonatkozó (3.3.4) állításunkat felhasználva ezt kapjuk:

$$\frac{1}{\varepsilon} \geq \frac{N}{2} \big(n!H(f)^{n-1} + 1 \big),$$

amelyből a szokásos módon, (3.3.2) segítségével tudjuk befejezni a bizonyítást. ■

3.3.6. Tétel. *Ha f egy egész együtthatós expanzív polinom és α egy gyöke, akkor*

$$\frac{1}{|\alpha| - 1} \leq \begin{cases} nL(f) + n & (\alpha \in \mathbb{R}), \\ 2\binom{n}{2} L(f)^{n-1} + 2\binom{n}{2} + 1 & (\alpha \in \mathbb{C} \setminus \mathbb{R}). \end{cases}$$

Bizonyítás. A bizonyítás megegyezik az előző tételével, kivéve az $|f^{(k)}(\pm 1)/k!|$ és az $|F^{(k)}(1)/k!|$ becslését. Az első (ami akkor kell, ha α valós) így módosul:

$$\left| \frac{f^{(k)}(\pm 1)}{k!} \right| = \left| \sum_{j=k}^{n} (\pm 1)^{j-k} \binom{j}{k} a_j \right| \leq \binom{n}{k} \sum_{j=k}^{n} |a_j| \leq \binom{n}{k} L(f).$$

A másodikhoz (ha α nem valós) elegendő $n!H(f)^{n-1}$-t lecserélni $L(f)^{n-1}$-re, azaz megmutatni, hogy $\sum_{l=0}^{N}|A_l| \le L(f)^{n-1}$. Tekintsük a (3.3.5) kifejtést. Vegyük észre, hogy a determináns minden oszlopában minden együttható legfeljebb egyszer szerepel, ezért a kifejtésben egyik $(s_{i,1},\, s_{i,2},\, \ldots,\, s_{i,n-1})$ indexsorozat sem fordulhat elő többször. Így ha ezen T_n darab sorozat helyett az összes lehetséges indexsorozatot vesszük $(1,1,\ldots,1)$-től $(n-1,n-1,\ldots,n-1)$-ig, akkor nem csökkenhet az érték:

$$|F(1)| \le \sum_{l=0}^{N}|A_l| \le \sum_{i=1}^{T_n}\prod_{j=1}^{n-1}|a_{s_{i,j}}| \le \sum_{i_1=1}^{n-1}\sum_{i_2=1}^{n-1}\cdots\sum_{i_{n-1}=1}^{n-1}\prod_{j=1}^{n-1}|a_{i_j}| = L(f)^{n-1}. \qquad \blacksquare$$

3.3.3. Korlátok összehasonlítása

Az alábbi táblázatban összehasonlítottuk az $\frac{1}{|\alpha|-1}$-re bebizonyított különféle korlátokat. Az elhanyagolható tagokat nem tüntettük fel.

3.3.7. Táblázat: Az expanzivitási hézag korlátjai

	$\alpha \in \mathbb{R}$	$\alpha \in \mathbb{C} \setminus \mathbb{R}$
3.3.3. Tétel	$2^{n-1}\lvert a_0\rvert$	$2^{\binom{n}{2}}\lvert a_0\rvert^{n-1}$
3.3.4. Tétel	$2^{n-2}(\lvert a_0\rvert + \lvert a_n\rvert)$	$2^{\binom{n-1}{2}}(\lvert a_0\rvert + \lvert a_n\rvert)^{n-1}$
3.3.5. Tétel	$\binom{n+1}{2}H(f)$	$\binom{n}{2}n!H(f)^{n-1}$
3.3.6. Tétel	$nL(f)$	$2\binom{n}{2}L(f)^{n-1}$

Látható, hogy a valós korlátok c_nA alakúak, a nem valósak pedig $c_n'A^{n-1}$ alakúak, és mindegyik esetben az utóbbi a nagyobb (legalábbis $n \ge 3$-ra).

Most összehasonlítjuk az egyes sorokat mindkét oszlopban. Először is,

$$|a_0| \le |a_0| + |a_n| \le L(f), \quad |a_0| + |a_n| \le 2H(f), \quad H(f) \le L(f),$$

vagyis egyik sor sem egyértelműen jobb (kisebb), mint bármelyik alatta lévő. Másodszor, mivel expanzív polinomokra $|a_n| < |a_0|$, ezért a második sor határozottan jobb, mint az első. Az $|a_0| + |a_n|$ és a $H(f)$ összehasonlítására az együtthatók közötti egyenlőtlenséget (3.1.4. Lemma) és a Stirling-approximációt ($\sqrt{2\pi n}\left(\frac{n}{e}\right)^n \le n! \le e\sqrt{n}\left(\frac{n}{e}\right)^n$) használjuk:

$$H(f) = \max_{k=0}^{n}|a_k| \le \max_{k=0}^{n}\left(\binom{n-1}{k-1}|a_n| + \binom{n-1}{k}|a_0|\right) \le$$
$$\le \binom{n-1}{\frac{n-1}{2}}(|a_n| + |a_0|) \le \frac{e}{\pi}\frac{2^{n-1}}{\sqrt{n-1}}(|a_n| + |a_0|).$$

Ebből látható, hogy a harmadik sor általában nem jobb, mint a második. Az alábbiakból látható az is, hogy a negyedik sem jobb a másodiknál:

$$L(f) = \sum_{k=0}^{n} |a_k| \le \sum_{k=0}^{n} \left(\binom{n-1}{k-1} |a_n| + \binom{n-1}{k} |a_0| \right) = 2^{n-1}(|a_n| + |a_0|).$$

Továbbá, mivel $L(f) \le (n+1)H(f)$, a negyedik sor a harmadiknál sem jobb általában.

A fentieket röviden úgy foglalhatjuk össze, hogy általában egyik sor sem jobb semelyik másiknál sem, kivéve, hogy a második sor jobb az elsőnél. Hogy melyiket érdemes használni, az a konkrét alkalmazástól függ. Például ha olyan polinomunk van, ahol a középső együtthatók jóval nagyobbak, mint a két szélső, akkor $|a_0| + |a_n|$ lehet a legjobb korlát, más esetekben viszont $H(f)$ vagy $L(f)$.

3.4. Alig expanzív polinomcsalád

Az előző szakaszban azt vizsgáltuk, hogy az expanzivitási hézagnak (3.3.1. Definíció) milyen elméleti alsó korlátjai lehetnek. Természetes módon merül fel a kérdés, hogy ezek a korlátok vajon javíthatók-e tovább, vagy élesek. Ebben a szakaszban ezt a kérdést részben megválaszoljuk. Az itteni eredmények nagyrészt a szerző következő publikációján alapulnak: [4].

Azt fogjuk látni, hogy rögzített n esetén a korlátjaink nagyságrendje éles. Más szóval az optimális alsó korlát is $1/c_n A$ alakú (valós esetben), ill. $1/c'_n A^{n-1}$ alakú (nem valós esetben) (vö. 3.3.7. Táblázat), legfeljebb a c_n és a c'_n szorzó javítható a korábbi eredményeinkhez képest. Ezt úgy fogjuk megmutatni, hogy konstruálunk olyan „alig expanzív" polinomokat, amelyeknek az expanzivitási hézagja megfelelő nagyságrendű.

Valós esetben könnyű a feladat. Legyen

$$f(x) := (A-1)x^n - A,$$

ennek az expanzivitási hézagja

$$\varepsilon = \sqrt[n]{1 + \frac{1}{A-1}} - 1 \approx \frac{1}{nA}.$$

A továbbiakban a nem valós esettel foglalkozunk. A szakasz hátralévő részében konstruktív bizonyítást adunk a következő tételre:

3.4.1. Tétel. *Minden $n \ge 2$-re és minden elég nagy A pozitív egész számra létezik olyan n-edfokú, A magasságú (3.1.2. Definíció) egész együtthatós expanzív polinom, amelynek az expanzivitási hézagja a következő nagyságrendű:*

$$\varepsilon = \frac{1}{2A^{n-1}} + O\left(\frac{1}{A^n}\right).$$

3.4.1. Motzkin-háromszög és inverze

A polinomkonstrukciónkhoz szükség lesz egy bizonyos számtáblázatra, az ún. Motzkin-háromszögre [27]. Ebben a szakaszban erről lesz szó.

Legyen $M_{n,k}$ $(n, k \geq 0)$ a $\mathbb{Z}^2$ rácson a $(0,0)$-ból az (n,k)-ba vezető utak száma, ha az $(1,1)$, $(1,0)$ és az $(1,-1)$ lépéseket használhatjuk, és az út sosem megy az x-tengely alá. Vegyük továbbá $(M_{n,k})$-nak, mint végtelen mátrixnak az inverzét, és jelöljük $(N_{n,k})$-val.

Az első néhány $M_{n,k}$- és $N_{n,k}$-érték a következő:

$M_{n,k}$	$k=0$	1	2	3	4	5	6	7
$n=0$	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0
2	2	2	1	0	0	0	0	0
3	4	5	3	1	0	0	0	0
4	9	12	9	4	1	0	0	0
5	21	30	25	14	5	1	0	0
6	51	76	69	44	20	6	1	0
7	127	196	189	133	70	27	7	1

$N_{n,k}$	$k=0$	1	2	3	4	5	6	7
$n=0$	1	0	0	0	0	0	0	0
1	-1	1	0	0	0	0	0	0
2	0	-2	1	0	0	0	0	0
3	1	1	-3	1	0	0	0	0
4	-1	2	3	-4	1	0	0	0
5	0	-4	2	6	-5	1	0	0
6	1	2	-9	0	10	-6	1	0
7	-1	3	9	-15	-5	15	-7	1

A definícióból is és a táblázatból is látszik, hogy ha $n < k$, akkor $M_{n,k} = N_{n,k} = 0$ (ezért Motzkin-*háromszög*), ha pedig $n = k$, akkor $M_{n,k} = N_{n,k} = 1$.

Az egyszerűbb számolás kedvéért célszerű bevezetni néhány kiegészítő értéket: legyen $M_{-1,-1} := 1$ és $M_{-1,k} = M_{n,-1} := 0$ $(k, n \geq 0)$, és legyen hasonlóan $N_{n,k}$-ra is. Könnyű látni, hogy az $M_{n,k}$-értékekre a következő rekurzív összefüggés áll fenn:

$$M_{n,k} = M_{n-1,k-1} + M_{n-1,k} + M_{n-1,k+1} \qquad (n, k \geq 0). \tag{3.4.1}$$

A Motzkin-háromszögnek és inverzének számos érdekes tulajdonsága van, nekünk ezek közül az alábbi összefüggésekre lesz szükségünk:

3.4.2. Lemma.

$$\sum_{k=i}^{j} M_{j,k} N_{k,i} = \delta_{i,j} \qquad (i, j \geq -1), \tag{3.4.2}$$

$$\sum_{k=i}^{j-l} M_{j,l+k} N_{k,i} = M_{j-i-1,l-1} \qquad (l \geq 0,\ -1 \leq i \leq j), \tag{3.4.3}$$

$$\sum_{k=0}^{\min(i,j)} M_{i,k} M_{j,k} = M_{i+j,0} \qquad (i, j \geq 0), \tag{3.4.4}$$

ahol $\delta_{i,j}$ a Kronecker-féle delta függvény, azaz $\delta_{i,i} = 1$, és $\delta_{i,j} = 0$ $(i \neq j)$.

Bizonyítás. A (3.4.2) egyszerűen csak azt jelenti, hogy N az M inverze.

A (3.4.3) lényegében a (3.4.2) általánosítása, és megegyezik vele $l = 0$-ra (ha $i \leq j$). Ha $l \geq 1$, akkor j szerinti indukcióval bizonyítunk. $j = i$-re az állítás triviális, a $j \to j+1$ lépésben pedig felhasználjuk a (3.4.1) rekurzív összefüggést:

$$\sum_{k=i}^{j-l+1} M_{j+1,l+k} N_{k,i} = \sum_{k=i}^{j-l+1} M_{j,l+k-1} N_{k,i} + \sum_{k=i}^{j-l} M_{j,l+k} N_{k,i} + \sum_{k=i}^{j-l-1} M_{j,l+k+1} N_{k,i} =$$
$$= M_{j-i-1,l-2} + M_{j-i-1,l-1} + M_{j-i-1,l} = M_{j-i,l-1}.$$

A (3.4.4)-et i-szerinti indukcióval bizonyítjuk rögzített $i + j$ mellett. Az összeg felső határaként írhatunk $\min(i, j)$ helyett nagyobbat, pl. i-t, mert a további értékek 0-k. Az $i = 0$ eset triviális, az $i \to i + 1$ lépés pedig a következőképpen bizonyítható:

$$\sum_{k=0}^{i+1} M_{i+1,k} M_{j-1,k} = \sum_{k=0}^{i+1} M_{i,k-1} M_{j-1,k} + \sum_{k=0}^{i+1} M_{i,k} M_{j-1,k} + \sum_{k=0}^{i+1} M_{i,k+1} M_{j-1,k} =$$
$$= \sum_{k=0}^{i} M_{i,k} M_{j-1,k+1} + \sum_{k=0}^{i} M_{i,k} M_{j-1,k} + \sum_{k=0}^{i} M_{i,k} M_{j-1,k-1} = \sum_{k=0}^{i} M_{i,k} M_{j,k}. \qquad \blacksquare$$

3.4.2. A polinomcsalád és tulajdonságai

Ebben a szakaszban definiáljuk azokat a polinomokat, amelyre a 3.4.1. Tételt bizonyítani fogjuk, és levezetjük néhány fontos tulajdonságukat. Minden $n \geq 2$ és $A \geq 1$ egész számra definiáljuk az $f(x)$ együtthatóit a következőképpen:

$$\begin{aligned} a_0 &:= A, \\ a_1 &:= A - (M_{n-3,0} + 1), \\ a_2 &:= A - M_{n-2,0}, \\ a_i &:= -M_{n-2,i-2} \qquad (3 \leq i \leq n). \end{aligned} \tag{3.4.5}$$

Az első néhány fokszámra így néznek ki ezek a polinomok:

$$\begin{aligned}
n = 2: &\quad (A-1)x^2 + (A-1)x + A, \\
n = 3: &\quad -x^3 + (A-1)x^2 + (A-2)x + A, \\
n = 4: &\quad -x^4 - 2x^3 + (A-2)x^2 + (A-2)x + A, \\
n = 5: &\quad -x^5 - 3x^4 - 5x^3 + (A-4)x^2 + (A-3)x + A, \\
n = 6: &\quad -x^6 - 4x^5 - 9x^4 - 12x^3 + (A-9)x^2 + (A-5)x + A, \\
n = 7: &\quad -x^7 - 5x^6 - 14x^5 - 25x^4 - 30x^3 + (A-21)x^2 + (A-10)x + A.
\end{aligned}$$

Érdekességként megjegyezzük, hogy az a_1 definíciójában szereplő 1-es helyett néha -1-et is írhatunk, pontosabban egy olyan ω-t, amelyre $\omega^{n-2} = 1$: $a_1 := A - (M_{n-3,0} + \omega)$. Ha n páratlan, akkor csak $\omega = 1$ lehet, de páros n-re lehet akár 1, akár -1. Sőt, ha $n = 2$, akkor ω tetszőleges egész szám lehet. A továbbiakban ezt az általánosított definíciót fogjuk használni a_1-re, és ω-ról csak annyit teszünk fel, hogy $\omega^{n-2} = 1$.

Az első lépés az, hogy bebizonyítsuk, hogy ezek a polinomok valóban expanzívak, legalábbis elég nagy A-ra. Ehhez az expanzivitás szükséges és elégséges feltételeiről szóló 3.2.2. Tételt fogjuk használni, és megvizsgáljuk a benne szereplő mennyiségeket: $f(\pm 1)$-et és $D_k^{\pm}(f)$-et. Ezzel kapcsolatban az alábbi tulajdonságokat tudjuk bizonyítani:

3.4.3. Lemma. *A fent definiált polinomokra a következők teljesülnek:*

1. $f(\pm 1) > 0$, *ha A elég nagy.*
2. *Minden k-ra ($1 \le k \le n-1$) és mindkét előjelre $D_k^{\pm}(f) > 0$ (3.2.1. Definíció), ha A elég nagy.*
3. $D_{n-1}^{-}(f) = 1$ *minden $A \in \mathbb{N}^+$-ra.*

Bizonyítás. Az első állítás triviális.

Jelölje a $D_k^{\pm}(f)$ determináns elemeit $d_{i,j}$ $(1 \le i, j \le k)$, ekkor a 3.2.1. Definíció és (3.4.5) alapján:

$$d_{i,j} = a_{j-i} \pm a_{i+j+n-k-1} =$$

$$= \mp M_{n-2,i+j+n-k-3} + \begin{cases} 0 & (j < i), \\ A \pm A & (j = i = 1 \wedge k = n-1), \\ A & (j = i, \text{ egyébként}), \\ A - M_{n-3,0} - \omega & (j = i+1), \\ A - M_{n-2,0} & (j = i+2), \\ -M_{n-2,j-i-2} & (j \ge i+3). \end{cases} \tag{3.4.6}$$

Például $n = 7$-re és $k = 6$-ra:

$$D_6^{-}(f) = \begin{vmatrix} 21 & A+20 & A+4 & -16 & -20 & -13 \\ 30 & A+25 & A+4 & A-16 & -29 & -25 \\ 25 & 14 & A+5 & A-9 & A-21 & -30 \\ 14 & 5 & 1 & A & A-10 & A-21 \\ 5 & 1 & & & A & A-10 \\ 1 & & & & & A \end{vmatrix}.$$

Ezek a determinánsok legfeljebb k-adfokú polinomok A-ban. Csak a főátlóban és a fölötte lévő két szomszédos átlóban szerepel az A, és minden előfordulásában 1 az együtthatója, kivéve a bal felső sarokban, ahol $k < n-1$ esetén 1, $k = n-1$ és „+" előjel esetén 2, $k = n-1$ és „−" előjel esetén pedig 0. Az első két esetben tehát a polinom főtagja A^k vagy $2A^k$, ami igazolja, hogy elég nagy A-ra a determináns pozitív. A bizonyítás hátralévő részében megmutatjuk, hogy a harmadik esetben is pozitív, egészen pontosan – és ez a lemma harmadik állítása – mindig 1.

Legyen D a $D^-_{n-1}(f)$ determináns mátrixa, és ezt fogjuk determinánstartó traszformációkkal olyan alakra hozni, amelyből már nyilvánvaló, hogy a determinánsa 1.

Első lépésben megszabadulunk a főátlón kívüli A-ktől. Informálisan úgy írhatjuk le ezt a transzformációt, hogy az utolsó sort kivonjuk a fölötte lévő kettőből, majd az utolsó előtti sort az afölötti kettőből, és így tovább lentről fölfelé. Formálisan ezt egy mátrixszorzással írjuk le: legyen $D = ED'$, ahol D'-ben csak főátlóban szerepel A, és E a transzformációs mátrix, amely a következőképpen néz ki: a főátlóban és a fölötte lévő két átlóban 1-esek vannak (mint ahol a D-ben A-k, de a bal felső sarokban is), a többi helyen pedig 0-k. Például $n = 6$-ra:

$$\begin{bmatrix} 9 & A+7 & A & -8 & -8 \\ 12 & A+9 & A-1 & A-8 & -12 \\ 9 & 4 & A+1 & A-5 & A-9 \\ 4 & 1 & 0 & A & A-5 \\ 1 & 0 & 0 & 0 & A \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 1 & 0 & -1 \\ 4 & A+5 & -2 & -3 & -3 \\ 5 & 3 & A+1 & -5 & -4 \\ 3 & 1 & 0 & A & -5 \\ 1 & 0 & 0 & 0 & A \end{bmatrix}.$$

A példából látszik, hogy D' struktúrája a D-éhez hasonló, de annál egyszerűbb. Ez általánosabban is igaz: az $M_{n,k}$-kra vonatkozó (3.4.1) rekurzió segítségével könnyű megmutatni, hogy (vö. (3.4.6))

$$d'_{i,j} = M_{n-3,i+j-3} + \begin{cases} 0 & (j < i), \\ 0 & (j = i = 1), \\ A & (j = i \geq 2), \\ -M_{n-3,j-i-1} - \omega N_{j-i-1,0} & (j > i). \end{cases} \tag{3.4.7}$$

Következő lépésként egy hasonlósági traszformációt hajtunk végre D'-n, ahol a hasonlósági mátrix lényegében a Motzkin-háromszög kezdőszelete. Egészen pontosan kiszámítjuk a $D'' := M^{-1}D'M$ szorzatot, ahol M elemei $m_{i,j} := M_{j-2,i-2}$, és M^{-1} elemei $n_{i,j} := N_{j-2,i-2}$. Az $n = 6$-os példát folytatva:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 1 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 & 1 & 0 & -1 \\ 4 & A{+}5 & -2 & -3 & -3 \\ 5 & 3 & A{+}1 & -5 & -4 \\ 3 & 1 & 0 & A & -5 \\ 1 & 0 & 0 & 0 & A \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 2 & 4 \\ 0 & 0 & 1 & 2 & 5 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 \\ 0 & A{+}2 & -1 & 0 & 0 \\ 0 & 1 & A{+}2 & -1 & 0 \\ 0 & 1 & 1 & A{+}2 & -1 \\ 1 & 0 & 0 & 0 & A \end{bmatrix}.$$

A szorzatot úgy számítjuk ki, hogy D'-t felbontjuk: $D' = U - V + W$, ahol U a bal felső háromszöges minta, V a jobb felső háromszög, és W-ben vannak az A-k. Egészen pontosan (vö. (3.4.7)):

$$\begin{aligned} u_{i,j} &= M_{n-3,i+j-3}, \\ v_{i,j} &= M_{n-3,j-i-1} + \omega N_{j-i-1,0} && (j > i), \\ w_{i,j} &= \delta_{i,j} A && (i \geq 2). \end{aligned}$$

Tagonként elvégezve a transzformációt:

$$(M^{-1}UM)_{i,j} = \sum_{l=1}^{\min(j,n-i)} \sum_{k=i}^{n-l} N_{k-2,i-2}\, M_{n-3,l+k-3}\, M_{j-2,l-2} =$$

$$\overset{(3.4.3)}{=} \sum_{l=1}^{\min(j,n-i)} M_{n-i-2,l-2}\, M_{j-2,l-2} \overset{(3.4.4)}{=} \begin{cases} \delta_{i,n-1} & (j = 1), \\ \delta_{j,1} & (i = n-1), \\ M_{n-i+j-4,0} & (\text{egyébként}). \end{cases}$$

$$\begin{aligned} (M^{-1}VM)_{i,j} &= \sum_{k=1}^{n-1} \sum_{l=k+1}^{n-1} N_{k-2,i-2}\, (M_{n-3,l-k-1} + \omega N_{l-k-1,0})\, M_{j-2,l-2} = \\ &= \sum_{m=1}^{n-2} (M_{n-3,m-1} + \omega N_{m-1,0}) \sum_{k=1}^{n-m-1} M_{j-2,m+k-2}\, N_{k-2,i-2} = \\ &\overset{(3.4.3)}{=} \sum_{m=1}^{j-i} (M_{n-3,m-1} + \omega N_{m-1,0})\, M_{j-i-1,m-1} = \\ &\overset{(3.4.4),(3.4.2)}{=} \begin{cases} M_{n-i+j-4,0} + \omega \delta_{i+1,j} & (j \geq i+1), \\ 0 & (\text{egyébként}). \end{cases} \end{aligned}$$

$$(M^{-1}WM)_{i,j} = \sum_{k=2}^{n-1} N_{k-2,i-2} A M_{j-2,k-2} \overset{(3.4.2)}{=} A \cdot \begin{cases} \delta_{i,j} & (i \geq 2), \\ 0 & (\text{egyébként}). \end{cases}$$

Ezeket összeillesztve a következőt kapjuk $D'' := M^{-1}D'M$-re (összhangban a fenti

példa jobb oldalával):

$$d''_{i,j} = (M^{-1}D'M)_{i,j} = \begin{cases} \delta_{i,n-1} & (j=1), \\ A\delta_{j,n-1} & (i=n-1 \wedge j \geq 2), \\ M_{n-i+j-4,0} + A\delta_{i,j} & (2 \leq j \leq i \leq n-2), \\ -\omega\delta_{i+1,j} & (j \geq i+1). \end{cases}$$

Ebből könnyű látni, hogy $\det D'' = \omega^{n-2} = 1$, tehát az eredeti determináns is 1. ■

3.4.3. A bizonyítás befejezése

Most már megvannak az eszközeink ahhoz, hogy bebizonyítsuk a 3.4.1. Tételt a (3.4.5)-ben definiált $f(x)$ polinomokra. Azt már tudjuk, hogy expanzívak (ha A elég nagy), mert a 3.4.3. Lemma éppen azt mondja, hogy az $f(x)$-ből képezhető minden mennyiség, ami a 3.2.2. Tétel alapján az expanzivitáshoz szükséges, valóban pozitív. Azt kell még megmutatnunk, hogy ezek a polinomok „éppen hogy" expanzívak, azaz a legkisebb gyökük közel van az egységkörhöz, mégpedig éppen a kívánt nagyságrendben. Ehhez megvizsgáljuk az $f_\varepsilon(x) := f((1+\varepsilon)x)$ polinomokat, és belőlük is képezzük ugyanazokat az expanzivitáshoz szükséges mennyiségeket. A legkisebb ε, amelyre valamelyik ilyen mennyiség nem lesz pozitív, tehát amelyre $f_\varepsilon(x)$ nem lesz expanzív, megadja az $f(x)$ legkisebb gyökének távolságát az egységkörtől.

Legyen $Q_0(A)$ a 3.2.2. Tétel bármelyik mennyisége (tehát $D_k^{\pm}(f)$ vagy $f(\pm 1)$), és legyen $Q(A,\varepsilon)$ ugyanez $f_\varepsilon(x)$-re. $Q(A,\varepsilon)$ úgy kapható meg $Q_0(A)$-ból, hogy lecseréljük az $a_0, a_1, \ldots, a_n$ együtthatókat a_0, $a_1(1+\varepsilon)$, …, $a_n(1+\varepsilon)^n$-re. Tekintsük $Q(A,\varepsilon)$-t egy kétváltozós polinomnak A-ban és ε-ban, és legyen $d := \deg_A Q$ és $N := \deg_\varepsilon Q$. Írjuk fel ε szerint csoportosítva:

$$Q(A,\varepsilon) = Q_0(A) + Q_1(A)\varepsilon + Q_2(A)\varepsilon^2 + \ldots + Q_N(A)\varepsilon^N,$$

ahol minden $\deg Q_i \leq d$.

Keressük azt a legkisebb ε-t minden elég nagy A-ra, amelyre valamelyik $Q(A,\varepsilon)$ nulla lesz. Először megmutatjuk, hogy ilyen ε létezik, és $\varepsilon = O(1/A)$.

Ha $\varepsilon = O(1/A)$, akkor a $Q(A,\varepsilon)$ fenti kifejtése felírható a következő formában, kihasználva, hogy $Q_i(A) = O(A^d)$:

$$Q(A,\varepsilon) = Q_0(A) + \big(Q_1(A) + O(A^{d-1})\big)\varepsilon. \tag{3.4.8}$$

Azokra a Q-kra, melyekre $\deg Q_0 = d$, azaz $Q_0(A) = cA^d + O(A^{d-1})$ valamely pozitív c-re (negatív nem lehet a 3.4.3. Lemma miatt), a (3.4.8) kifejtés így írható fel:

$$Q(A, \varepsilon) = cA^d + O(A^{d-1}),$$

és ez pozitív, ha A elég nagy. A 3.4.3. Lemma bizonyításának eleje alapján ez az eset a következő Q-kra áll fenn:

1. $Q = f(\pm 1)$, ekkor $d = 1$ és $c = 2 \pm 1$;
2. $Q = D_k^{\pm}(f)$ $(k < n-1)$, ekkor $d = k$ és $c = 1$;
3. $Q = D_{n-1}^{+}(f)$, ekkor $d = n-1$ és $c = 2$.

Ami maradt, az a $Q = D_{n-1}^{-}(f)$, ahol $Q_0(A) = 1$ (ugyancsak a 3.4.3. Lemma alapján). Megmutatjuk, hogy ekkor $Q(A, \varepsilon)$ következő polinomja teljes fokszámú: $\deg Q_1 = n-1$. Ehhez megvizsgáljuk az A^{n-1} együtthatóját a $Q(A, \varepsilon)$-nak az A szerinti kifejtésében. $Q(A, \varepsilon)$ struktúrája hasonló, mint $D_{n-1}^{-}(f)$-é (lásd (3.4.6)-ot), de az a_k együtthatókat lecseréltük $a_k(1+\varepsilon)^k$-ra. Továbbra is egy $(n-1) \times (n-1)$-es determinánsról van szó, amelyben az A továbbra is csak a felső háromszög részben szerepel, és mindenütt csak első fokon, így A^{n-1} csak a főátlóbeli elemek szorzatából állhat elő. Ezek az elemek:

$$q_{k,k} = a_0 - a_{2k}(1+\varepsilon)^{2k} = \begin{cases} A + M_{n-2,2k-2}(1+\varepsilon)^{2k} & (k \geq 2), \\ (-2\varepsilon - \varepsilon^2)A + M_{n-2,0}(1+\varepsilon)^2 & (k = 1). \end{cases}$$

Ebből kijön, hogy A^{n-1} együtthatója $-2\varepsilon - \varepsilon^2$, következésképp Q_1 főtagja $-2A^{n-1}$.

Helyettesítsük be ezt és a $Q_0(A) = 1$-et a (3.4.8) kifejtésbe:

$$Q(A, \varepsilon) = 1 + \big(-2A^{n-1} + O(A^{n-2})\big)\varepsilon. \tag{3.4.9}$$

Helyettesítsünk be két konkrét ε-t:

$$Q(A, 0) = 1,$$
$$Q(A, 1/A) = 1 - 2A^{n-2} + O(A^{n-3}).$$

Az első pozitív, a második pedig negatív, ha A elég nagy, ezért kell hogy legyen olyan $\varepsilon = O(1/A)$, ahol $Q(A, \varepsilon) = 0$. A pontos nagyságrendjét úgy találhatjuk meg, hogy a (3.4.9)-et nullával tesszük egyenlővé, és átrendezzük:

$$\varepsilon = \frac{1}{2A^{n-1} + O(A^{n-2})} = \frac{1}{2A^{n-1}} + O\left(\frac{1}{A^n}\right).$$

Mivel a fentiek szerint az összes többi Q pozitív, ha $\varepsilon = O(1/A)$ (és A elég nagy), ezért ez lesz a legelső ε, amelyre a 3.2.2. Tétel valamelyik feltétele nem teljesül. Ezzel befejeztük a 3.4.1. Tétel bizonyítását. ■

3.5. Általánosított számrendszerek eldöntési algoritmusának futási ideje

Ebben a szakaszban bemutatunk egy számítást, amely expanzív polinomokat használ, és amelyben alkalmazni tudjuk néhány eddigi eredményünket. A számrendszerfogalom egy bizonyos általánosításáról lesz szó, az ún. mátrix alapú számrendszerekről. Ahhoz, hogy egy mátrixból számrendszert lehessen képezni, szükséges feltétel, hogy a mátrix karakterisztikus polinomja expanzív legyen. Egy olyan algoritmust vizsgálunk meg, amely eldönti, hogy egy mátrix adott számjegyekkel számrendszert alkot-e. Mint látni fogjuk, ez az algoritmus annál lassabb, minél kevésbé teljesül az expanzivitás, így a futási idő korlátozásához szükségünk van az expanzivitási hézag egy alsó korlátjára.

Ez a szakasz a szerző másokkal közös [5] publikációján alapul. Az első részben (3.5.1. szakasz) röviden bevezetjük a témakör fontosabb fogalmait és állításait. Ezekről részletesebben van szó például az [5, 10, 19, 21] publikációkban, valamint az ezekben hivatkozottakban. Ezután (3.5.2. és 3.5.3. szakasz) a szerző saját számításai következnek. Az [5] többi szerzőjének eredményeit (például a véletlenített algoritmusokról) a dolgozatban nem mutatjuk be.

3.5.1. Definíciók és előzmények

A *számrendszer* hétköznapi értelemben a következőt jelenti: adott egy $q \geq 2$ egész szám (*alapszám*), ekkor minden $x \in \mathbb{Z}^+$ egyértelműen felírható az alábbi formában:

$$x = d_0 + d_1 q + d_2 q^2 + \ldots + d_k q^k,$$

ahol $d_0, \ldots, d_k \in \{0, 1, \ldots, q-1\}$, és $d_k \neq 0$.

Ezt a fogalmat általánosítjuk mátrixokra és vektorokra n dimenzióban.

3.5.1. Definíció. Legyen $M \in \mathbb{Z}^{n \times n}$ és $D \subset \mathbb{Z}^n$. Az (M, D) párost *általánosított számrendszer*nek nevezzük, ha minden $x \in \mathbb{Z}^n$ egyértelműen felírható az alábbi formában:

$$x = \sum_{j=0}^{k} M^j d_j,$$

ahol $k \geq 0$, $d_j \in D$ és $d_k \neq 0$, ha $k > 0$. Ekkor M-et a számrendszer *alapjá*nak, D elemeit pedig *számjegy*eknek nevezzük.

Az alábbiakban bemutatjuk az általánosított számrendszerek néhány alapvető tulaj-

donságát, de ezekhez először bevezetünk egy-két fogalmat.

3.5.2. Definíció. Az $M \in \mathbb{R}^{n\times n}$ mátrix *expanzív*, ha a karakterisztikus polinomja expanzív, azaz ha minden sajátértéke az egységkörön kívül található.

A következő állítás magyarázza meg az „expanzív" elnevezést:

3.5.3. Lemma. *Ha $M \in \mathbb{R}^{n\times n}$ egy expanzív mátrix, akkor van olyan $c > 1$ konstans és $\|\cdot\|$ norma, hogy minden $x \in \mathbb{R}^n$-re $\|Mx\| \geq c\|x\|$.*

3.5.4. Definíció. Legyen $x, y \in \mathbb{Z}^n$ és $M \in \mathbb{Z}^{n\times n}$, ekkor $x \equiv y \mod M$ (x és y *kongruens* modulo M), ha van olyan $z \in \mathbb{Z}^n$, hogy $x - y = Mz$.

3.5.5. Definíció. Egy $X \subset \mathbb{Z}^n$ halmaz *teljes maradékrendszer* modulo M, ha bármely különböző $x, y \in X$-re $x \not\equiv y \mod M$, valamint bármely $x \in \mathbb{Z}^n$-re van olyan $y \in X$, hogy $x \equiv y \mod M$.

Megjegyzés: az első feltétel miatt a második y egyértelműen létezik.

3.5.6. Lemma. *Ha M invertálható, és X teljes maradékrendszer modulo M, akkor X elemszáma $|\det M|$.*

3.5.7. Tétel. *Ha (M, D) általánosított számrendszer, akkor*

1. *M invertálható,*
2. *M expanzív,*
3. *$0 \in D$, és*
4. *D teljes maradékrendszer modulo M.*

A bizonyítás nemtriviális részei megtalálhatók például a következő helyeken: [21, 33].

Mivel ezeket a feltételeket konkrét esetben (viszonylag) könnyű ellenőrizni, ezért a továbbiakban csak ilyen (M, D)-kkel foglalkozunk.

A számrendszerekre a következő ekvivalens megfogalmazást lehet adni:

3.5.8. Tétel. *Legyen $M \in \mathbb{Z}^{n\times n}$ invertálható és $0 \in D \subset \mathbb{Z}^n$ teljes maradékrendszer, valamint legyen $\tau\colon \mathbb{Z}^n \to \mathbb{Z}^n$:*

$$\tau(x) := M^{-1}(x - d) \qquad (d \in D,\ x \equiv d \mod M).$$

Ekkor (M, D) akkor és csak akkor általánosított számrendszer, ha minden $x \in \mathbb{Z}^n$-re az x, $\tau(x)$, $\tau^2(x) = \tau(\tau(x))$, ... sorozatban szerepel a 0.

A τ definíciójának van értelme, mert d egyértelműen létezik a teljes maradékrendszer miatt. Számrendszer esetén a τ jelentése az, hogy az x vektor (M, D) szerinti felírásának levágja az utolsó számjegyét. Ha egyszer eljutottunk a 0-ba, akkor a sorozat onnantól kezdve végig 0.

Ha szeretnénk eldönteni, hogy egy adott (M, D) számrendszert alkot-e, a fenti tételt közvetlenül nem tudjuk alkalmazni, hiszen végtelen sok $x \in \mathbb{Z}^n$ vektort kéne ellenőrizni. Az expanzivitás miatt azonban erre nincs szükség:

3.5.9. Tétel. *Legyen M, D és τ az előző tétel szerint. Ha M expanzív, akkor létezik olyan véges $H \subset \mathbb{Z}^n$ halmaz, hogy bármely $x \in \mathbb{Z}^n$ esetén az $x, \tau(x), \tau^2(x), \ldots$ sorozat előbb-utóbb csak H-beli elemeket tartalmaz.*

Ez már lehetővé teszi a következő eldöntési algoritmust: határozzunk meg egy ilyen H halmazt, és minden $x \in H$-ra kezdjük el kiszámítani az $x, \tau(x), \tau^2(x), \ldots$ sorozatot. Ha minden esetben eljutunk a 0-ba, akkor (M, D) számrendszer. Mivel H véges, ezért minden sorozat előbb-utóbb ismétlődik, így véges sok tagot kell csak kiszámítani.

Az algoritmus futási ideje természetesen H méretétől függ. A továbbiakban ezzel fogunk foglalkozni, felhasználva néhány eredményünket az expanzivitási hézagról.

3.5.2. A véges halmaz mérete

Ebben a szakaszban bebizonyítjuk a véges H halmaz létezéséről szóló 3.5.9. Tételt úgy, hogy egyúttal kiszámítunk egy konkrét ilyen H-t a mátrix és a számjegyhalmaz paramétereiből.

Szükségünk lesz a következő fogalomra:

3.5.10. Definíció. Egy $A \in \mathbb{R}^{n \times n}$ mátrix *Jordan-kondíciószám*a a következő mennyiség:

$$K(A) := \inf_T \|T\| \, \|T^{-1}\|,$$

ahol $A = T^{-1} J T$ az A egy Jordan-felbontása, az infímum rögzített J mellett az összes lehetséges T-re vonatkozik, és $\|\cdot\|$ tetszőleges rögzített természetes mátrixnorma (vagyis amit vektornorma indukált).

A Jordan-kondíciószám nem tévesztendő össze a közönséges kondíciószámmal, amely egyszerűen $\|A\| \, \|A^{-1}\|$. Ha A egész mátrix, és dimenziója és elemeinek korlátja rögzített, akkor a Jordan-kondíciószámnak van egy véges felső korlátja, hiszen csak véges sok ilyen mátrix létezik. Szebb lenne explicit képletet adni erre a felső korlátra a dimenzió és az elemkorlát függvényében, de ez egyelőre nem sikerült, így az eredményben a Jordan-kondíciószám is szerepelni fog.

A H halmaz létezéséhez fontos feltétel, hogy M expanzív. Emiatt a $\rho(M^{-1})$ spektrálsugár (azaz M^{-1} legnagyobb sajátértékének abszolút értéke) kisebb, mint 1. Következésképp – a spektrálsugár tulajdonságai miatt – létezik olyan $\|\cdot\|$ norma, melyre

$\rho(M^{-1}) \le \|M^{-1}\| < 1$. Legyen $B' := \max_{d\in D} \|d\|/(1-\|M^{-1}\|)$, ekkor

$$\|\tau(x)\| = \|M^{-1}(x-d)\| \le \|M^{-1}\|(\|x\|+\|d\|) \le \|M^{-1}\|\big(\|x\| + (1-\|M^{-1}\|)B'\big) \le$$
$$\le \|M^{-1}\|(2-\|M^{-1}\|)\max(\|x\|, B') = \big(1-(1-\|M^{-1}\|)^2\big)\max(\|x\|, B').$$

A $\max(\|x\|, B')$ szorzójára, jelöljük q-val, teljesül, hogy $0 < q < 1$. A fenti számítás alapján ha $\|x\| \ge B'$, akkor $\|\tau(x)\| \le q\|x\|$, ha pedig $\|x\| \le B'$, akkor $\|\tau(x)\| \le B'$. Ezekből következik, hogy az x, $\tau(x)$, $\tau^2(x)$, ... sorozat elemei előbb-utóbb bekerülnek a $H = \{x \in \mathbb{Z}^n \mid \|x\| \le B'\}$ halmazba.

Ezzel ugyan bebizonyítottuk a tételt, de még nem vagyunk készen: a megadott H nem elég konkrét, mert a $\|\cdot\|$ normát nem ismerjük, csak annyit tudunk róla, hogy $\|M^{-1}\| < 1$. A cél az, hogy egy könnyen kezelhető norma szerepeljen a képletben, például $\|x\|_1 := |x_1| + |x_2| + \ldots + |x_n|$.

Ismeretes, hogy $\mathbb{R}^n$-ben bármely két norma ekvivalens, azaz van olyan c_1 és c_2 pozitív szám, hogy $c_1\|x\|_1 \le \|x\| \le c_2\|x\|_1$. Ebből és a B' definíciójából könnyen látható, hogy $\|x\| \le B' \implies \|x\|_1 \le B$, ha B a következő (vagy nagyobb) érték:

$$B = \frac{c_2}{c_1} \cdot \frac{\max\limits_{d\in D} \|d\|_1}{1-\|M^{-1}\|}. \tag{3.5.1}$$

Ahhoz, hogy kiszámítsuk a c_1 és c_2 konstansokat, meg kell konstruálni egy megfelelő normát, amelyre $\|M^{-1}\| < 1$. Írjuk M^{-1}-et a következő alakban: $M^{-1} = T^{-1}JT$, ahol J az M^{-1} Jordan-normálformája. Legyen $0 < \delta < 1$, és legyen J_δ olyan, mint J, csak az átlón kívüli 1-esek helyett δ van. Ez felírható $J_\delta = D_\delta J D_\delta^{-1}$ alakban, ahol D_δ egy δ-hatványokat tartalmazó diagonális mátrix. Ezen mátrixok struktúráját a következő példával illusztráljuk:

$$J = \left(\begin{array}{cccc|ccc|c}
\frac{1}{\lambda_1} & 1 & & & & & & \\
 & \frac{1}{\lambda_1} & 1 & & & & & \\
 & & \frac{1}{\lambda_1} & 1 & & & & \\
 & & & \frac{1}{\lambda_1} & & & & \\
\hline
 & & & & \frac{1}{\lambda_2} & 1 & & \\
 & & & & & \frac{1}{\lambda_2} & 1 & \\
 & & & & & & \frac{1}{\lambda_2} & \\
\hline
 & & & & & & & \frac{1}{\lambda_3}
\end{array}\right), \quad
J_\delta = \left(\begin{array}{cccc|ccc|c}
\frac{1}{\lambda_1} & \delta & & & & & & \\
 & \frac{1}{\lambda_1} & \delta & & & & & \\
 & & \frac{1}{\lambda_1} & \delta & & & & \\
 & & & \frac{1}{\lambda_1} & & & & \\
\hline
 & & & & \frac{1}{\lambda_2} & \delta & & \\
 & & & & & \frac{1}{\lambda_2} & \delta & \\
 & & & & & & \frac{1}{\lambda_2} & \\
\hline
 & & & & & & & \frac{1}{\lambda_3}
\end{array}\right), \quad
D_\delta = \left(\begin{array}{cccc|ccc|c}
\delta^{-1} & & & & & & & \\
 & \delta^{-2} & & & & & & \\
 & & \delta^{-3} & & & & & \\
 & & & \delta^{-4} & & & & \\
\hline
 & & & & \delta^{-1} & & & \\
 & & & & & \delta^{-2} & & \\
 & & & & & & \delta^{-3} & \\
\hline
 & & & & & & & \delta^{-1}
\end{array}\right).$$

Ezután a normát a következőképpen definiáljuk: $\|x\|_\delta := \|D_\delta T x\|_1$, ebből az $\|A\|_\delta := \|D_\delta T A T^{-1} D_\delta^{-1}\|_1$ mátrixnorma indukálható. Ha $\delta < 1 - \rho(M^{-1})$, akkor ez a norma megfelel a követelményeinknek:

$$\|M^{-1}\|_\delta = \|D_\delta T M^{-1} T^{-1} D_\delta^{-1}\|_1 = \|J_\delta\|_1 \le \rho(M^{-1}) + \delta < 1.$$

Szükségünk van még az 1-es és a δ-norma közötti ekvivalencia konstansaira. Könnyű megmutatni, hogy

$$\delta^{-1}\|T^{-1}\|_1^{-1}\|x\|_1 \leq \|x\|_\delta \leq \delta^{-m}\|T\|_1\|x\|_1,$$

ahol m a legnagyobb Jordan-block mérete J-ben. Ha ezeket a konstansokat behelyettesítjük (3.5.1)-be, és felhasználjuk, hogy $\|M^{-1}\|_\delta \leq \rho(M^{-1}) + \delta$, akkor azt kapjuk, hogy a következő (vagy nagyobb) B megfelelő:

$$B = \frac{\|T\|_1\|T^{-1}\|_1 \max\limits_{d\in D}\|d\|_1}{\delta^{m-1}(1-\rho(M^{-1})-\delta)}. \tag{3.5.2}$$

Ez a B bármely T-re működik, amelyre $M^{-1} = T^{-1}J\,T$, következésképp $\inf_T B$ szintén megfelelő lesz. Ehhez elegendő észrevenni, hogy $\|\tau^k(x)\|_1$ mindig egész szám, tehát bármely B helyett írhatunk $\lfloor B\rfloor$-t, és így az infímum leegyszerűsödik minimumra. A $\|T\|_1\|T^{-1}\|_1$ infímuma pedig nem más, mint $K_1(M^{-1})$, az M^{-1} Jordan-kondíciószáma (3.5.10. Definíció).

A következő lépés a $\rho(M^{-1})$-re adni egy felső korlátot. Jelölje egy A mátrix sajátértékeit $\lambda_1(A)$, $\lambda_2(A)$, $\ldots$, $\lambda_n(A)$, ekkor $\rho(M^{-1}) = \max_i|\lambda_i(M^{-1})| = 1/\min_i|\lambda_i(M)| = 1/(1+\varepsilon)$, ahol ε az M karakterisztikus polinomjának expanzivitási hézagja (3.3.1. Definíció). Erre kiszámítottunk különböző alsó korlátokat a 3.3. szakaszban, amelyek összefoglalása a 3.3.7. Táblázatban található. Ezek közül a 3.3.3. Tételt fogjuk használni, mert jelen esetben a konstans tagnak kitüntetett szerepe van: egy karakterisztikus polinom konstans tagja a mátrix determinánsa, azt pedig tudjuk a 3.5.6. Lemma alapján, hogy $|\det M| = |D|$. A 3.3.3. Tételből következik, hogy $\varepsilon \geq 1/(N+1)$, ahol $N := 2^{\binom{n}{2}}|D|^{n-1}$. Emiatt $\rho(M^{-1}) = 1/(1+\varepsilon) \leq 1 - 1/(N+2)$, és ezzel csökkenteni tudjuk B nevezőjét (3.5.2)-ről az alábbira:

$$B = \frac{K_1(M^{-1})\max\limits_{d\in D}\|d\|_1}{\delta^{m-1}\left(\frac{1}{N+2}-\delta\right)}. \tag{3.5.3}$$

Utolsó lépésként egy δ-t kell választani $0 < \delta < 1/(N+2)$ között, célszerűen úgy, hogy B minél kisebb legyen. Ha $m = 1$ (azaz ha M diagonalizálható), akkor a minél kisebb δ a jó, sőt, vehetjük a $\delta \to 0$ infímumot (hasonló érveléssel, mint a (3.5.2) után a T-re), és a következőt kapjuk:

$$B = (N+2)K_1(M^{-1})\max_{d\in D}\|d\|_1. \tag{3.5.4}$$

Ha $m > 1$, akkor könnyen kiszámítható, hogy B (3.5.3)-beli nevezőjének maximuma a $\delta = \frac{m-1}{m(N+2)}$-nél van, ahol értéke $\frac{(1-1/m)^{m-1}}{m(N+2)^m}$. Az $(1-1/m)^{m-1}$ sorozatot lecserélhetjük az

infímumára, $1/e$-re, és m helyére n-et írhatunk (egyik lépés sem csökkenti B-t), ekkor a következő értéket kapjuk:

$$B = en(N+2)^n K_1(M^{-1}) \max_{d \in D} \|d\|_1. \tag{3.5.5}$$

A két képlet, (3.5.4) és (3.5.5) közül az utóbbi a nagyobb, ezért használhatjuk azt általánosan is. Behelyettesítve N-et, azt kapjuk végeredményként, hogy ha

$$B = en\Big(2^{\binom{n}{2}}|D|^{n-1} + 2\Big)^n K_1(M^{-1}) \max_{d \in D} \|d\|_1, \tag{3.5.6}$$

akkor a $H = \{x \in \mathbb{Z}^n \mid \|x\|_1 \leq B\}$ halmaz megfelel a 3.5.9. Tételnek.

3.5.3. A futási idő

Az eldöntési algoritmus a H halmaz minden pontjára iterálja τ-t, és ha mindegyik eljut a 0-ba, akkor (M, D) számrendszer, ha pedig valamelyik sorozatban 0-n kívüli ismétlődés van, akkor nem. Mivel a 3.5.9. Tétel miatt mindig H-ban maradunk, ezért legfeljebb $|H|$ elemet kell bejárni, hogy biztosan ismétlődést találjunk. Ez a gyakorlatban egy nagyon pesszimista korlát, de általánosan ennél jobbat nem tudunk mondani. Így az algoritmus futási ideje legfeljebb $|H|^2\,\mathrm{T}(\tau)$, ahol $\mathrm{T}(\tau)$, a τ kiértékelések futási ideje, a mátrixszorzás miatt $O(n^2)$. Ezen megállapítások részleteiért és az algoritmus különböző változataiért lásd [5]-öt.

A $H = \{x \in \mathbb{Z}^n \mid \|x\|_1 \leq B\}$ halmaz elemszámát az $\{x \in \mathbb{R}^n \mid \|x\|_1 \leq B\}$ alakzat n-dimenziós térfogatával közelítjük. Ez az alakzat az oktaéder n-dimenziós általánosítása („cross-polytope”), melynek térfogata $2^n B^n/n!$. Felhasználva B képletét, (3.5.6)-ot:

$$\begin{aligned} |H| &= O\left(\frac{2^n B^n}{n!}\right) = O\left(\frac{(2e)^n B^n}{n^n}\right) = \\ &= O\left((2e^2)^n\, 2^{\frac{n^3(n-1)}{2}}\, |D|^{n^3} K_1(M^{-1})^n \max_{d \in D} \|d\|_1^n\right). \end{aligned}$$

Az algoritmus futási idejére így a következő felső korlátot tudjuk adni:

$$\begin{aligned} T = O(|H|^2 n^2) &= O\left(n^2\,(2e^2)^{2n}\, 2^{n^3(n-1)}\, |D|^{2n^3} K_1(M^{-1})^{2n} \max_{d \in D} \|d\|_1^{2n}\right) = \\ &= O\left(2^{n^4} |D|^{2n^3} K_1(M^{-1})^{2n} \max_{d \in D} \|d\|_1^{2n}\right). \end{aligned}$$

Ez a nagyságrend erősen exponenciális, ami egyrészt mutatja az eldöntési probléma nehézségét, de ahogy azt a fentiekben írtuk, ez egy meglehetősen pesszimista becslés, és ilyen általános feltételek mellett nehéz ennél jobbat adni.

3.6. További kutatási lehetőségek

Ahány kérdést megválaszoltunk a 3. fejezetben, legalább annyi maradt még nyitva ebben a témakörben. Röviden felsorolunk néhány ilyet, ami szorosan kapcsolódik a tárgyalt eredményekhez, és ami a jövőbeni kutatás alapja lehet.

A 3.3. szakaszban különféle alsó korlátokat számítottunk ki az expanzivitási hézagra, ezek komplex gyökök esetén $1/c_n A^{n-1}$ alakúak (lásd a 3.3.7. táblázatot), a 3.4. szakaszban pedig olyan polinomokat definiáltunk, amelyek ε-ja $1/c'_n A^{n-1}$ nagyságrendű (lásd a 3.4.1. Tételt és a (3.4.5) konstrukciót). A két nagyságrend csak A-ban azonos, az n-től függő szorzó (c_n és c'_n) különböző. Jó lenne a két értéket közelebb hozni egymáshoz, vagy akár egyenlővé tenni őket, azaz megtalálni a lehető legjobb alsó korlátot az expanzivitási hézagra, és ennek élességét egy polinomcsaládd al bizonyítani.

Az expanzivitás D-feltételeiről szóló 3.2.2. Tétel kihasználta, hogy a polinom együtthatói valósak (lásd a tétel bizonyítása utáni megjegyzést). A jelen dolgozatban, ahol amúgyis egész együtthatós polinomokkal foglalkoztunk, ez bőven elegendő volt, de érdekes lenne általánosítani a tételt komplex együtthatókra is. Például ha az együtthatók Gauss-egészek, akkor ugyanúgy felmerül az együtthatórobbanás kérdése, mint egész együtthatók esetén, így egy hasonló tétel szintén hasznos lenne.

A 3.2.3. szakaszban definiált D-polinomokból $2n$-féle van (minden $1 \leq k \leq n$ közötti k-ra és mindkét előjelre), de ezek közül eddig csak a következő háromnak sikerült a szerkezetét megfejteni: $\widetilde{D}^-_{n-1}(f)(x)$, valamint a kapcsolódó $\widetilde{D}^+_n(f)(x)$ és $\widetilde{D}^-_n(f)(x)$ (lásd a 3.2.7. Lemmát). Érdekes lenne megvizsgálni a többit is, és megfejteni például azt, hogy a gyökeik hogyan viszonyulnak f gyökeihez.

A 3.3.5. Tételben az ε-ra kiszámított korlátban szerepel $n!$, amely a $D^-_{n-1}(f)$ kifejtésében szereplő tagok számának, T_n-nek egy felső korlátja. Ez nem a legjobb korlát, ezért jó lenne alaposabban megvizsgálni T_n-t, és javítani $n!$-t, vagy akár zárt formulát adni T_n-re. A sorozat első néhány értéke $n = 1, 2, \ldots$-re $T_n = 1, 2, 4, 12, 40, \ldots$

Az általánosított számrendszerek futási idejének kiszámításában fontos szerephez jutott a Jordan-kondíciószám (3.5.10. Definíció), és a végeredményt is ennek függvényében írtuk fel. Jó lenne, ha ezt a komplikált fogalmat ki tudnánk küszöbölni a képletekből, azaz jó lenne $K_1(M^{-1})$-et kifejezni a mátrix paramétereinek függvényében, mint a dimenzió és az elemek mérete.

Összefoglalás

A doktori értekezés algebrai számokkal kapcsolatos különféle algoritmusokról és ezekhez kapcsolódó elméleti kérdésekről szól. A vizsgált algoritmusok fontos jellemzője, hogy nem numerikusan, hanem egzaktul számolnak. Ezzel elkerüljük a numerikus számolásokra jellemző kerekítési hibákat, viszont az egzakt algoritmusok általában lassabbak, és felléphet az ún. együtthatórobbanás. Ezért a futási idő kordában tartása (polinomiális idő garantálása) fontos szempont az egzakt algoritmusok tervezése során.

A disszertáció két nagy témakörre bontható. Az első felében algebrai számokon végzett műveletek és algoritmusok futási idejét vizsgáljuk. Egy-két korábbi eredményt is felhasználva kiszámítjuk a fontosabb műveletek hatékonyságát algebrai számtestekben, köztük olyanokét is, mint a valós algebrai számok összehasonlítása és az egészre kerekítés. Ezekre támaszkodva megvizsgáljuk két algoritmusnak, a Bareiss-algoritmusnak (amely a Gauss-elimináció módosítása) és az LLL-algoritmusnak (amely egy rácsredukciós algoritmus) a futási idejét, ha egzakt algebrai számokkal végezzük őket. Egész számokra mindkét algoritmusnak ismert a viselkedése, itt kiterjesztjük őket algebrai egészekre, és bebizonyítjuk, hogy ebben az esetben is polinomiális a futási idejük.

A dolgozat második felében speciális algebrai számokról van szó: olyan egész együtthatós polinomok gyökeiről, amelyek minden gyöke a komplex egységkörön kívül van. Az ilyen polinomokat expanzív polinomoknak nevezzük. Foglalkozunk azzal a kérdéssel, hogy hogyan tudjuk eldönteni egy polinomról, hogy expanzív-e (a gyökök kiszámítása nélkül). Erre léteznek módszerek, de itt bevezetünk egy új algoritmust, amely garantáltan nem vezet együtthatórobbanáshoz. Ezután megvizsgáljuk azt a kérdést, hogy egész együtthatós expanzív polinomok gyökei mennyire lehetnek közel az egységkörhöz. Erre kiszámítunk különféle alsó korlátokat a polinom paramétereinek függvényében, valamint konstruálunk egy expanzív polinomcsaládot, amelynek vannak az egységkörhöz nagyon közeli gyökei, és ezáltal megmutatjuk, hogy az alsó korlátok közel élesek.

Summary in English

The dissertation is about algorithms of algebraic numbers and related theoretical questions. An important aspect of the discussed algorithms is that they calculate exactly, not numerically. In this way we avoid the usual problem of numerical calculations, the rounding errors, but the exact algorithms are usually slower, and they often suffer from coefficient explosion. Avoiding this problem and maintaining the running time (i.e. ensuring polynomial time) is an important aspect of exact algorithms.

The dissertation has two major parts. The first half deals with the computational complexity of operations and algorithms involving algebraic numbers. We calculate the running time of the main operations in algebraic number fields (using some existing results). We also analyse some special operations like comparison between real algebraic numbers and integer rounding. These results enable us to analyse two algorithms, the Bareiss algorithm (a generalisation of the Gaussian elimination) and the LLL algorithm (a lattice reduction algorithm) on algebraic numbers using exact arithmetic, and analyse their complexity. Their behaviour is well-known for integers, but we extend them to algebraic integers, and prove that their running time remains polynomial.

The second half of the dissertation is about special algebraic numbers: roots of integer polynomials whose all roots are outside of the complex unit circle. Such polynomials are called expansive polinomials. We examine the question how to decide whether a polynomial with integer coefficients is expansive (without calculating its roots). There are some existing methods for this, but we introduce a new method that is guaranteed to avoid coefficient explosion. Then we examine how close the roots of an expansive integer polynomial can be to the unit circle, and give lower bounds on this distance in terms of the parameters of the polynomial. Then we construct a family of expansive polynomials that have roots very close to the unit circle, thus showing that the lower bounds are almost sharp.

Publikációk

[1] Uray M.J.: Szimbolikus számítások algebrai számokkal. *MSc diplomamunka*, Eötvös Loránd Tudományegyetem, Budapest, 2016

[2] Uray M.J.: Algebraic number fields and the LLL algorithm. *Elbírálás alatt*, kézirat: `https://arxiv.org/abs/1810.01634`

[3] Uray M.J.: Characterization of expansive polynomials by special determinants. *Publ. Math. Debrecen* **98/3-4** (2021), 379–399. o.

[4] Uray M.J.: A family of barely expansive polynomials. *Annales Univ. Sci. Budapest., Sect. Math.*, elfogadott

[5] Bóka D., Burcsi P., Uray M.J.: On the algorithmic complexity of some numeration-related problems. *Publ. Math. Debrecen* **98/1-2** (2021), 65–81. o.

[6] Uray M.J.: On proving inequalities by cylindrical algebraic decomposition. *Annales Univ. Sci. Budapest., Sect. Comp.* **51** (2020), 231–252. o.

Irodalomjegyzék

[7] Bareiss, E.H.: Sylvester's identity and multistep integer-preserving Gaussian elimination. *Mathematics of Computation* **22** (1968), 565–578. o.

[8] Belabas, K.: Topics in computational algebraic number theory. *Journal de Théorie des Nombres de Bordeaux* **16** (2004), 19–63. o.

[9] Biasse, J.-F., Fieker, C., Hofmann, T.: On the computation of the HNF of a module over the ring of integers of a number field. *Journal of Symbolic Computation* **80** (2017), 581–615. o.

[10] Burcsi P.: Algorithmic aspects of generalized number systems. *PhD értekezés*, Eötvös Loránd Tudományegyetem, Budapest, 2008

[11] Burcsi P., Kovács A.: An algorithm checking a necessary condition of number system constructions. *Annales Univ. Sci. Budapest., Sect. Comp.* **25** (2005), 143–152. o.

[12] Burcsi P., Kovács A.: Exhaustive search methods for CNS polynomials, *Monatshefte für Mathematik* **155** (2008), 421–430. o.

[13] Cassels, J.W.S.: An Introduction to the Geometry of Numbers. *Springer-Verlag Berlin Heidelberg*, 1997

[14] Cohen, H.: A Course in Computational Algebraic Number Theory. *Springer-Verlag Berlin Heidelberg*, 1996

[15] Cohn, A.: Über die Anzahl der Wurzeln einer algebraischen Gleichung in einem Kreise. *Mathematische Zeitschrift* **14** (1922), 110–148. o.

[16] Dixon, J.D.: Exact solution of linear equations using p-adic expansions. *Numerische Mathematik* **40** (1982), 137–141. o.

[17] Fieker, C., Stehlé, D.: Short bases of lattices over number fields. *Algorithmic Number Theory, 9th International Symposium, Proceedings* **6197** (2010), 157–173. o.

[18] Geddes, K.O., Czapor, S.R., Labahn, G.: Algorithms for Computer Algebra. *Kluwer Academic Publishers*, 1992

[19] Kátai I.: Generalized number systems in Euclidean spaces. *Mathematical and Computer Modelling* **38** (2003), 883–892. o.

[20] Knuth, D.E.: The Art of Computer Programming, Seminumerical Algorithms, Vol. 2. *Addison-Wesley*, 1997

[21] Kovács A.: On number expansions in lattices. *Mathematical and Computer Modelling* **38** (2003), 909–915. o.

[22] Lee, C., Pellet-Mary, A., Stehlé, D., Wallet, A.: An LLL algorithm for module lattices. *Advances in Cryptology, ASIACRYPT* (2019), 59–90. o.

[23] Lehmer, D.H.: A machine method for solving polynomial equations. *Journal of the ACM* **8** (1961), 151–162. o.

[24] Lehmer, D.H.: Factorization of certain cyclotomic functions, *Annals of Mathematics* **34** (1933), 461–479. o.

[25] Lenstra, A.K., Lenstra, H.W., Lovász, L.: Factoring Polynomials with Rational Coefficients. *Mathematische Annalen* **261** (1982), 515–534. o.

[26] Morel, I., Stehlé D., Villard G.: H-LLL: Using Householder inside LLL. *Proceedings of ISSAC* (2009), 271–278. o.

[27] The On-Line Encyclopedia of Integer Sequences, 2020, A026300 sorozat, `http://oeis.org/A026300`

[28] Nguyen, P.Q., Stehlé, D.: An LLL algorithm with quadratic complexity. *SIAM Journal on Computing* **39** (2009), 874–903. o.

[29] Pethő A., Pohst, M.E., Bertók, Cs.: On multidimensional Diophantine approximation of algebraic numbers. *Journal of Number Theory* **171** (2017), 422–448. o.

[30] Plantard, T., Susilo, W., Zhang, Z.: Adaptive precision floating point LLL. *Information Security and Privacy, ACISP 2013, Lecture Notes in Computer Science* **7959** (2013). 104–117. o.

[31] Pohst, M., Zassenhaus, H.: Algorithmic Algebraic Number Theory. *Cambridge University Press*, 1993

[32] Saruchi, G., Morel, I., Stehlé, D., Villard, G.: LLL reducing with the most significant bits. *ISSAC, Kobe, Japan* (2014), 367–374. o.

[33] Vince, A.: Replicating tesselations. *SIAM Journal on Discrete Mathematics* **6** (1993), 501–521. o.

[34] Waldschmidt, M.: Diophantine Approximation on Linear Algebraic Groups, Transcendence Properties of the Exponential Function in Several Variables, Grundlehren der Mathematischen Wissenschaften. *Springer-Verlag Berlin Heidelberg*, 2000