Lineáris egyenletrendszer kerekítési hibák nélküli megoldása moduláris algoritmusok segítségével

Erdélyi Balázs

„A matematikában a gondolat, ami számít."
Szonja Vasziljevna Kovalevszkaja

I. Bevezetés

A nagy számokkal végzendő elemi műveleteket nemcsak elméleti érdeklődésből vizsgálhatjuk, mivel sok olyan eljárás létezik, melyben az egyes lépések során végrehajtott kerekítések végül az eredményt teljesen használhatatlanná teszik. A moduláris algoritmusok, néhány egyszerű számelméleti tény felhasználásával, alkalmasak olyan algebrai feladatok megoldására, ahol a részeredmények jegyeinek száma nagyra nőhet, kerekíteni pedig nem szabad. Cikkünkben olyan eljárást mutatunk meg, melynek segítségével a lineáris egyenletrendszer megoldása során a kerekítési hibák nélküli számolás még akkor is megvalósítható, ha a bemenő adatok hibáin már nem áll módunkban változtatni.

II. Hibák és algoritmusok

Tekintsük az alábbi példát.

Oldjuk meg az alábbi lineáris egyenletrendszert!

$$\begin{aligned} a_{11}x_1 + \cdots + a_{1n}x_n &= b_1 \\ &\vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{1}$$

Jól megfogalmazottnak nevezhetjük a feladatot?

Bizonyos esetekben nem, hiszen előfordulhat, hogy a megoldás nem egyértelmű, illetve nincs is megoldás. Helyesebb akkor talán úgy megfogalmazni a kérdést, hogy megoldható-e (1), és ha igen, akkor írjuk fel a maximális számú független megoldást? Sajnos nem. A kérdésünket még így sem nevezhetjük értelmesnek, mivel a rendszer bemenő adatai – az A együtthatómátrix és a b vektor – a gyakorlati alkalmazások során rendszerint csak közelítően ismertek, ahol

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \quad \text{és} \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}.$$

Márpedig, ha a pontos értékük nem ismert, és a megoldás nem egyértelmű, akkor a b vektor akármilyen kis változtatása olyan egyenlethez vezet, melynek nincs megoldása. (Megjegyezzük, hogy ha A-t változtatjuk meg kismértékben, akkor az új egyenletnek majdnem mindig pontosan egy megoldása lesz.) Ha azonban a megoldás egyértelmű, akkor is előfordulhat, hogy a bemenő adatok jelentéktelen (pontossági határokon belül eső) változtatása a megoldás óriási arányú változásához vezet. (Bahvalov példája a VI. szakszaban)

Tehát célunk nem pontos megoldás megadása, hanem jó algoritmusok keresése közelítő megoldások kiszámítására.

III. A hiba forrása, stabilitás

A hiba forrása sokféle lehet. Beszélhetünk kerekítési hibáról, amely abból adódik, hogy a numerikus adatok tárolási formája a számítógépeknél (fix, illetve lebegőpontos esetben is) csupán véges pontosságú. Az egészek halmaza például megszámlálhatóan végtelen számosságú, viszont a számítógép csak véges sok egész tárolására képes.

Előfordulhat, hogy a kerekítési hiba a számítás során megnő, az eredményben dominánssá válik, így nem stabilis eljárást kapunk.

Létezik olyan hibafajta is, melyet program határoz meg, mint például a csonkítási hiba. Lényege, hogy a legtöbb numerikus módszer diszkrét módon közelít valamilyen folytonos mennyiséget. Egy integrált például úgy számítunk ki numerikusan, hogy kiszámítjuk a függvény értékét véges sok pontban, de korántsem minden pontban. Egy másik példa, amikor egy függvény értékét számítjuk ki valamely a függvényt közelítő sor első néhány tagjának összegeként, de természetesen sohasem veszünk végtelen sok tagot. A csonkítási hiba még akkor is meglenne, ha a számítógépünk abszolút pontossággal tárolná a számokat, és végezné a műveleteket. A numerikus analízisnek ezért egyik fő feladata az ilyen típusú hibák minimalizálása.

A kerekítési és csonkítási hiba általában független egymástól, így szükséges egy olyan eljárás definiálása, mely lehetővé teszi, hogy ha a lineáris egyenletrendszer megoldása során a bemenő adatok hibáin már nem is áll módunkban változtatni, a kerekítési hibák nélküli számolás mégis megvalósítható legyen.

IV. Szükséges ismeretek

Az (1) megoldási eljárása során szükséges ismeretek a következők.

a) *Cramer-szabály*: ha egy lineáris egyenletrendszer szabályos, akkor megoldható, és pontosan egy megoldása van, amit a következő alakban keresünk (ahol D_i-t D-ből úgy kapjuk, hogy a D i-edik oszlopának elemei helyére a b oszlopvektor elemeit helyettesítjük):

$$x_i = \frac{D_i}{D} = \frac{\begin{vmatrix} a_{11} & \dots & \overset{\downarrow}{b_1} & \dots & a_{1n} \\ & & \vdots & & \\ a_{1n} & \dots & b_1 & \dots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & \dots & a_{1n} \\ & \vdots & \\ a_{1n} & \dots & a_{nn} \end{vmatrix}} \qquad D \neq 0,\ i = 1, \dots, n.$$

(i-dik oszlop)

b) *Kínai maradéktétel* (Sun Tsu Suan-Ching – i. sz. 350): Ha az $m_1, m_2, \dots, m_k$ modulusok páronként relatív prímek, akkor a

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ &\vdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

kongruenciarendszernek bármely $a_1, \dots, a_k \in \mathbb{Z}$ esetén van megoldása, és ez a megoldás modulo $m_1, m_2, \dots, m_k$ egyértelműen meghatározott.

(Megjegyzés: $x < m_1 m_2 \cdots m_k$.)

c) A moduláris algoritmusok alkalmazása során lényeges, hogy egy determináns abszolút értékét egyszerű módon tudjuk felülről becsülni, mivel az egészértékű determináns meghatározásában a közbülső számolás során előforduló értékek a moduláris eljárások során mindig egy adott korlát alatt maradnak. Erre alkalmas a következő tétel.

Hadamard-egyenlőtlenség: $D \leq \prod_{i=1}^{n} \sqrt{a_{1i}^2 + \dots + a_{ni}^2}$.

Ebből következik, hogy ha bármely $a_{ij} < M \ \Rightarrow\ D \leq \left(\sqrt{n}M\right)^n$.

Az eredménynek egy másik következménye, hogy ha

$$a_{ij}, b_i \leq M \ (i, j = 1, \dots, n) \ \Rightarrow\ D_i \leq \left(\sqrt{n}M\right)^n.$$

V. Az eljárás lépései

Tekintsük az (1) bemenő adatait, a_{ij} és b_i $(i, j = 1, \ldots, n)$ számokat egészeknek. (Ha nem lennének azok, akkor közös nevezőjükkel beszorozva egésszé tehetők.) Mivel az eljárás megoldásait a Cramer-szabály alkalmazásával kapjuk meg, így feladatunk a $D, D_1, \ldots D_n \in \mathbb{Z}$ meghatározása.

1. Első lépésként szükségünk van egy prímszámtáblázatra, amely sok, M-nél kisebb, de ezen belül minél nagyobb prímszámot tartalmaz. A táblázatból kiválasztunk egy p_1 prímet, kiszámoljuk D legkisebb maradékát p_1-el osztva, amit jelöljünk $D^{(1)}$-el. Előfordulhat, hogy $D^{(1)} = 0$, ekkor dobjuk el p_1-et, mivel számunkra nem használható. Ebben az esetben p_1-et *szerencsétlen prímszám*nak nevezzük. Annak valószínűsége, hogy ilyen prímet találunk, igen kicsi, mivel éppen D prímosztóiról van szó, márpedig p_1 legfeljebb $1/p_1$ valószínűséggel lesz osztója D-nek. Ha $D^{(1)} \neq 0$, akkor vegyük a következő prímszámot és ismételjük az előbbi eljárást mindaddig, míg végül rendelkezésünkre állnak a $p_1, p_2, \ldots, p_S$ prímszámok, melyekre D legkisebb maradékának értéke $D^{(1)}, \ldots, D^{(S)}$, melyek egyike sem egyenlő nullával.

Fontos kérdés, hogy hány darab prímszámot kell vennünk, azaz mekkora az s értéke? Annyi prímszámra lesz szükségünk, míg nem teljesül az alábbi egyenlőtlenség:

$$p_1 p_2 \cdots p_S > \left(\sqrt{n}M\right)^n.$$

Az alábbi táblázatban összefoglaltunk két példát.

n	M	$\left(\sqrt{n}M\right)^n$	M-nél kisebb prímek szozata	s
2	100	20000	716539	3
10	100	$1E+25$	$1,15E+25$	14

A lineáris egyenletrendszerekről jelen esetben tehát annyit tudunk, hogy az együtthatók mindegyike kisebb, mint 100. Az első esetben az ismeretlenek száma 2, a másodikban 10. Ilyen feltételek mellett az algoritmus első lépése három, illetve 14 prímszám kiválasztását követően fog véget érni.

Ha $n = 10$, akkor a lehetséges kiválasztandó prímek és nem megfelelőségi valószínűségük a következő:

M-nél kisebb prímek	37	41	43	47	53	59	61	67	71	73	79	83	89	97
Szerencsétlen prím valószínűsége (%)	2.70	2.44	2.33	2.13	1.89	1.69	1.64	1.49	1.41	1.37	1.27	1.20	1.12	1.03

A szerencsétlen prímszám találati valószínűségéből látható, hogy miért szükséges minél nagyobb prímeket választani (melyek M-nél kisebbek).

2. Az algoritmus második lépéseként, oldjuk meg az

$$Ax \equiv D^{(1)}b \pmod{p_1} \tag{2}$$

kongruenciarendszert. A megoldás egyértelmű lesz, mivel csak olyan eseteket tartottunk meg, amelyekben a D maradéka p_1-el osztva nem nulla. A Cramer-szabályból tudjuk, hogy a $D_1, \dots D_n \in \mathbb{Z}$ számok kielégítik az

$$\begin{aligned} a_{11}D_1 + \cdots + a_{1n}D_n &= Db_1 \\ &\vdots \\ a_{n1}D_1 + \cdots + a_{nn}D_n &= Db_n \end{aligned}$$

egyenletrendszert, így a (2) kongruenciarendszer megoldásai: $D_1^{(1)}, \dots, D_n^{(1)}$.

Az eljárást hasonlóan elvégezve a többi prímszámra, az alábbi megoldásokat kapjuk:

$$\left(D_1^{(1)}, \dots, D_n^{(1)}\right), \dots, \left(D_1^{(S)}, \dots, D_n^{(S)}\right).$$

3. A harmadik lépésben a kínai maradéktétel segítségével számíthatjuk ki azokat a legkisebb abszolút értékű $L, v_1, \dots, v_n$ számokat, melyekre fennáll, hogy

$$\begin{aligned} L \equiv D^{(1)},\ v_1 \equiv D_1^{(1)}, \dots, v_n \equiv D_n^{(1)} \pmod{p_1} \\ \vdots \\ L \equiv D^{(S)},\ v_1 \equiv D_1^{(S)}, \dots, v_n \equiv D_n^{(S)} \pmod{p_S}. \end{aligned}$$

Egyszerűen belátható, hogy $L = D, v_1 = D_1, \dots, v_n = D_n$, így a keresett számokat sikerült az algoritmus segítségével véges lépésben meghatároznunk.

A bemutatott módszer tehát lehető teszi, hogy egy lineáris egyenletrendszert kerekítési hibák nélkül úgy oldjunk meg, hogy a számítás menete közben esetleg fellépő, akár több száz jegyű számokkal nem kell dolgoznunk.

VI. Bahvalov példája

Tekintsük az alábbi A mátrixot, mely esetén a megoldás egyértelmű, mivel determinánsának értéke egy.

$$A = \begin{bmatrix} 1 & 0 & \dots & \dots & 0 \\ 2 & 1 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & 2 & 1 & 0 \\ 0 & \dots & 0 & 2 & 1 \end{bmatrix}$$

Oldjuk meg az (1) egyenletrendszert!

Behelyettesítve:

$$\begin{aligned} x_1 &= b_1 \\ 2x_1 + x_2 &= b_2 \\ &\vdots \\ 2x_{n-1} + x_n &= b_n \end{aligned}$$

A formális megoldást egyszerűen megkapjuk:

$$\begin{aligned} x_1 &= b_1 \\ x_2 &= b_2 - 2x_1 = b_2 - 2b_1 \\ &\vdots \\ x_n &= b_n - 2b_{n+1} + 4b_{n-2} - \ldots + (-2^{n-1})b_1 \\ &= (-2^{n-2})(-2b_1 + b_2) + (-2^{n-4})(-2b_3 + b_4) + \cdots \end{aligned}$$

Tegyük fel, hogy az egyenletrendszer jobb oldali konstansait csak közelítőleg ismerjük

$$b_{2n-1} \approx 0,5 \quad \text{és} \quad b_{2n} \approx 1 \quad (n \in \mathbb{N}^+).$$

Ekkor

$$x_n \approx \begin{cases} 0, & \text{ha } n \text{ páros}, \\ 1/2, & \text{ha } n \text{ páratlan}. \end{cases}$$

Mivel x_n egy viszonylag kis szám, ezért nem várunk nagy eltérést a végeredményben amiatt, mert b_n értékét nem tudjuk teljes pontossággal.

Márpedig, ha például b_1 értékét 10^{-7} pontossággal ismerjük és feltételezzük, hogy 100 darab ismeretlenünk van, akkor x_n hibája $2^{99} \cdot 10^{-7} \approx 6,34 \cdot 10^{22}$ nagyságú.

Irodalom

[1] Gács Péter, Lovász László, *Algoritmusok*, Tankönyvkiadó, 1989.
[2] Itshak Borosh, Aviezri S. Fraenkel, *Exact solutions of linear equations with rational coefficients by congruence techniques,* Mathematics of Computation, Providence, 1966, 107-112.
[3] Járai Antal, Kovács Attila, *Komputeralgebra,* Eötvös Kiadó, 2004.
[4] John Barkley Rosser, A method of computing exact inverses of matrices with integer coefficients, *Journal of Research*, Washington, 1952, 349-358.
[5] Lovász László, *Algoritmusok bonyolultsága*, ELTE, 2009.
[6] Nyikolaj Szergejevics Bahvalov, *A gépi matematika numerikus módszerei,* Műszaki Kiadó, 1977.
[7] Stoyan Gisbert, Takó Galina, *Numerikus módszerek I-III.*, Typotex, 2005.
[8] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, *Új algoritmusok*, Scolar, 2003.

Erdélyi Balázs
Erdelyi.Balazs@ejf.hu