Gauss-Jordan elimináció

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2020 október 5.

Feladat: megoldani az alábbi egyenletrendszert:

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

ahol **A** négyzetes mátrix.

- legegyszerűbb általános módszer lineáris egyenletrendszerek megoldására
- numerikusan legalább annyira stabil, mint más eljárás,
- főleg teljes *pivotolás*[1] esetén

[1] pivot = csuklópont, de a *pivoting* és *pivot element* szavaknak nincsen bevett fordítása

A Gauss–Jordan-elimináció tulajdonságai

Nagy egyenletrenszerek esetén fontos szempontok:

Memóriaigény:

- tárolni kell a mátrixot, és a megoldásvektort: $N^2 + N$
- vagy invertálás esetén a mátrixot és az inverzét: $2N^2$
- a mátrix inverzét elvileg lehet tárolni a bemeneti mátrix helyén

Számítási igény:

- szükséges műveletek száma $O(N^3)$
- nem a leggyorsabb módszer

Viszont egyszerű, sok mindent meg lehet érteni rajta keresztül

A megoldási módszer elemi lépései

A következő műveletek nem változtatják meg egy lineáris egyenletrendszer megoldását:

- *két sor felcserélése*
  ez nyilvánvaló, hiszen az egyenletek sorrendje teljesen tetszőleges, feltéve persze, hogy a jobboldal megfelelő sorait is megcseréljük
- *más sorok lineárkombinációjának hozzáadása bármely sorhoz*
  ebbe belefér az is, hogy egy sort számmal szorzunk, ha az a szám nem nulla; természetesen mindkét oldalon el kell végezni
- *a változók felcserélése*
  ez lényegileg nem változtat az egyenleteken, ha emlékszünk, hogy a végén a változókat megfelelő permutáció szerint vissza kell cserélgetni; ez az $\mathbf{A}$ mátrix oszlopainak felcserélését jelenti.

Kiindulás: kibővített mátrix

A megoldandó egyenletrendszer:

$$\begin{pmatrix} 2 & 6 & 2 \\ 1 & 1 & -1 \\ 3 & 9 & 5 \end{pmatrix} \cdot \mathbf{x} = \begin{pmatrix} 18 \\ 1 \\ 35 \end{pmatrix}$$

Felírjuk a mátrixot és a jobb oldalt a következő alakban:

$$\left(\begin{array}{rrr|r} 2 & 6 & 2 & 18 \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right)$$

Ha az inverz mátrixot keressük, akkor pedig:

$$\left(\begin{array}{rrr|rrr} 2 & 6 & 2 & 1 & 0 & 0 \\ 1 & 1 & -1 & 0 & 1 & 0 \\ 3 & 9 & 5 & 0 & 0 & 1 \end{array}\right)$$

- a gyakorlatban a jobb oldalt külön tömbben érdemes tárolni
- implementáláskor ne másoljuk egybe a két mátrixot!
- a jobb oldalon akárhány oszlop állhat, az egyenletrendszert egyszerre több **b** vektorra is megoldhatjuk

Cél: a korábban ismertetett elemi műveletek segítségével

- a baloldalt egységmátrix alakra hozni
- de közben a műveleteket a jobb oldalon is elvégezzük
- jobboldal effektive beszorozzuk az együttható mátrix inverzével

Az elimináció menete

Cél: a korábban ismertetett elemi műveletek segítségével

- a baloldalt egységmátrix alakra hozni
- de közben a műveleteket a jobb oldalon is elvégezzük
- jobboldal effektive beszorozzuk az együttható mátrix inverzével

Elindulunk a főátló első elemétől.

1. Ha az elem nem 1, akkor az egész sort elosztjuk a főátlóbeli elemmel, hogy a főátlóban 1 legyen

$$\left(\begin{array}{rrr|r} \mathbf{2} & 6 & 2 & 18 \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right) \longrightarrow \left(\begin{array}{rrr|r} \mathbf{1} & \mathbf{3} & \mathbf{1} & \mathbf{9} \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right)$$

2. Minden alatta levő sorból kivonjuk az első sor valahányszorosát úgy, hogy a főátló alatt végig 0 legyen

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ \mathbf{0} & -2 & -2 & -8 \\ \mathbf{0} & 0 & 2 & 8 \end{array}\right)$$

2. Minden alatta levő sorból kivonjuk az első sor valahányszorosát úgy, hogy a főátló alatt végig 0 legyen

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ \mathbf{0} & -2 & -2 & -8 \\ \mathbf{0} & 0 & 2 & 8 \end{array}\right)$$

3. Ha főátlóbeli elem alatt mindent kinulláztunk, akkor folytatjuk a főátló következő elemével:

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & \mathbf{-2} & -2 & -8 \\ 0 & 0 & 2 & 8 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & 4 \\ 0 & 0 & 2 & 8 \end{array}\right)$$

2. Minden alatta levő sorból kivonjuk az első sor valahányszorosát úgy, hogy a főátló alatt végig 0 legyen

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 1 & 1 & -1 & 1 \\ 3 & 9 & 5 & 35 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ \mathbf{0} & -2 & -2 & -8 \\ \mathbf{0} & 0 & 2 & 8 \end{array}\right)$$

3. Ha főátlóbeli elem alatt mindent kinulláztunk, akkor folytatjuk a főátló következő elemével:

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & \mathbf{-2} & -2 & -8 \\ 0 & 0 & 2 & 8 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & 4 \\ 0 & 0 & 2 & 8 \end{array}\right)$$

4. Majd kivonjuk a sor valahányszorosát az összes többiből úgy, hogy a főátlót kivéve mindenütt 0 legyen:

$$\left(\begin{array}{ccc|c} 1 & 3 & 1 & 9 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & 2 & 8 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & \mathbf{0} & -2 & -3 \\ 0 & 1 & 1 & 4 \\ 0 & \mathbf{0} & 2 & 8 \end{array}\right)$$

5. Az eljárást tovább folytatjuk a főátló elemeire

$$\left(\begin{array}{ccc|c} 1 & 0 & -2 & -3 \\ 0 & 1 & 1 & 4 \\ 0 & 0 & \mathbf{2} & 8 \end{array}\right) \longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & -2 & -3 \\ 0 & 1 & 1 & 4 \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{4} \end{array}\right)$$

$$\longrightarrow \left(\begin{array}{ccc|c} 1 & 0 & \mathbf{0} & 5 \\ 0 & 1 & \mathbf{0} & 0 \\ 0 & 0 & 1 & 4 \end{array}\right)$$

6. Az eljárás akkor ér végét, ha a baloldali részmátrix az egységmátrix alakját veszi fel. Ekkor jobboldalon vagy a megoldást kapjuk, vagy pedig a mátrix inverzét, attól függően miből indultunk ki.

$$\left(\begin{array}{ccc|c} \mathbf{1} & 0 & 0 & 5 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 4 \end{array}\right)$$

Lehetséges problémák

- zérus elem a főátlóban

$$\left(\begin{array}{cccc|c} 1 & 0 & 5 & & \\ 0 & 1 & 6 & & \\ 0 & 0 & \mathbf{0} & \cdots & \cdots \\ & & \vdots & & \end{array}\right)$$

Ilyenkor az adott sorral nem tudjuk az alatta és fölötte levő elemeket eliminálni.

Lehetséges problémák

- zérus elem a főátlóban

$$\left(\begin{array}{cccc|c} 1 & 0 & 5 & & \\ 0 & 1 & 6 & & \\ 0 & 0 & \mathbf{\color{red}0} & \cdots & \cdots \\ & & \vdots & & \end{array}\right)$$

Ilyenkor az adott sorral nem tudjuk az alatta és fölötte levő elemeket eliminálni.

Megoldás: sorcsere!
Keressünk a főátló aktuális is eleme alatti valamelyik sorban (de ugyanabban az oszlopban) olyan elemet, ami nem nulla (ún. *pivot-elem*), cseréljük meg a két sort (a jobboldalt is!), majd folytassuk az eliminációt $\rightarrow$ az egyenletek sorrendje tetszőleges !

$\Rightarrow$ *részleges pivotolás*.

Lehetséges problémák

- nagyon kis elem a van főátlóban
- ekkor az eliminációs lépésekben nagy szorzótényezők jelennek meg
- ez numerikus instabilitási problémákhoz vezethet

Megoldás: válasszuk az adott oszlop főátló alatti abszolút értékben legnagyobb elemét, a megfelelő sorokat cseréljük meg, és folytassuk így az eliminációt.

Az egyenletek átskálázása

A sorokat tetszőleges számmal szorozva bármelyik aktuális oszlopbeli, főátló alatti elem lehet maximális. Hogyan válasszuk ki azt az elemet, amellyel a pivotálást végezzük ?

Az egyenletek átskálázása

A sorokat tetszőleges számmal szorozva bármelyik aktuális oszlopbeli, főátló alatti elem lehet maximális. Hogyan válasszuk ki azt az elemet, amellyel a pivotálást végezzük ?

Megoldás: Úgy keressük a legnagyobb elemet, hogy a sorokat az eredeti mátrix sorainak legnagyobb elemével normáljuk.

- vagy a mátrix sorait a legelején leosztjuk minden sor abszolút értékben maximális elemével,
- vagy csak eltároljuk a legnagyobb elemeket és a döntésnél használjuk őket, de nem normáljuk explicit a sorokat

⇒*implicit pivotálás*

- az oszlop főátló alatti része csupa 0

$$\left(\begin{array}{cccc|c}
1 & 0 & 5 & & \\
0 & 1 & 6 & & \\
0 & 0 & \mathbf{0} & \cdots & \cdots \\
 & & \mathbf{0} & & \\
 & & \mathbf{0} & & \\
 & & \vdots & & \\
 & & \mathbf{0} & &
\end{array}\right)$$

Ez azt jelenti, hogy a mátrix szinguláris, nincs megoldás.

További lehetőség: oszlopcsere

Eddig csak az sorok cseréjéről volt szó.

Általánosabban: a főátlóbeli aktuális elem alatti, és tőle jobbra levő almátrixban keressük az abszolút értékben legnagyobb elemet, ez lesz a *pivot-elem*.

$$\left(\begin{array}{ccccc|c} 1 & 0 & 5 & & & \\ 0 & 1 & 6 & & & \\ 0 & 0 & \mathbf{0} & \cdots & \cdots & \cdots \\ & & & x & x & \\ & & & x & x & \end{array}\right)$$

Az x-szel jelölt elemek között keresünk új pivot elemet.

További lehetőség: oszlopcsere

Eddig csak az sorok cseréjéről volt szó.

Általánosabban: a főátlóbeli aktuális elem alatti, és tőle jobbra levő almátrixban keressük az abszolút értékben legnagyobb elemet, ez lesz a *pivot-elem*.

$$\left(\begin{array}{ccccc|c} 1 & 0 & 5 & & & \\ 0 & 1 & 6 & & & \\ 0 & 0 & \mathbf{0} & \cdots & \cdots & \cdots \\ & & & x & x & \\ & & & x & x & \end{array}\right)$$

Az x-szel jelölt elemek között keresünk új pivot elemet.

Tapasztalat:

- az algoritmus stabil, ha pivotnak mindig az almátrix abszolút értékben legnagyobb elemét választjuk.

További lehetőség: oszlopcsere

Eljárás:

A baloldali mátrix esetében:

- a megfelelő sorokat **és** oszlopokat megcseréljük, hogy a pivotelem a főátlóba kerüljön
- feljegyezzük, hogy melyik két oszlopot cseréltük meg
- ez valójában a változók átnevezése, így a jobboldal megfelelő sorait az eljárás végén vissza kell majd cserélni!

További lehetőség: oszlopcsere

Eljárás:

A baloldali mátrix esetében:

- a megfelelő sorokat **és** oszlopokat megcseréljük, hogy a pivotelem a főátlóba kerüljön
- feljegyezzük, hogy melyik két oszlopot cseréltük meg
- ez valójában a változók átnevezése, így a jobboldal megfelelő sorait az eljárás végén vissza kell majd cserélni!

A jobboldali mátrix esetében:

- csak a sorcserét végezzük el.

Majd mindkét oldalon az aktuális sorral lefelé eliminálunk.

Ezt nevezzük *teljes pivotolásnak*.

Melyiket használjuk?

- részleges pivotálás egyszerűbb, nem kell a megoldásvektor elemeinek permutálásával foglalkozni
- de csak olyan számok szolgálhatnak pivot elemként, amelyek már a megfelelő oszlopban vannak.

Általában a részleges pivotálás “majdnem” olyan jó, mint a teljes

Az algoritmus építőkövei

Azonosítjuk a fő részalgoritmusokat, amiket külön-külön könnyű megírni.

- Eldöntjük, hogy milyen formában tároljuk a mátrixokat (sorok vagy oszlopok szerint)
- Azonosítjuk az alapvető műveleteket:
    - Legnagyobb abszolút értékű elem megtalálása sorban
    - Sor szorzása (osztása) számmal
    - Két sor különbségének képzése
    - Legnagyobb abszolút értékű elem megtalálása oszlopban, főátló alatt
    - Két sor cseréje
    - Legnagyobb abszolút értékű elem megtalálása almátrixban
    - Két oszlop cseréje (oszlopcsere könyvelése)

Azonosítjuk a fő részalgoritmusokat, amiket külön-külön könnyű megírni.

- Eldöntjük, hogy milyen formában tároljuk a mátrixokat (sorok vagy oszlopok szerint)
- Azonosítjuk az alapvető műveleteket:
  - Legnagyobb abszolút értékű elem megtalálása sorban
  - Sor szorzása (osztása) számmal
  - Két sor különbségének képzése
  - Legnagyobb abszolút értékű elem megtalálása oszlopban, főátló alatt
  - Két sor cseréje
  - Legnagyobb abszolút értékű elem megtalálása almátrixban
  - Két oszlop cseréje (oszlopcsere könyvelése)

- Az építőelemeket külön-külön megvalósítjuk és *teszteljük*!
- Alulról felfelé, az egyes függvényeket alaposan kipróbálva haladjunk!

- Ezek után jöhet a fő algoritmus leprogramozása
- Többfajta mátrixon teszteljük (0 elem a főátlóban, stb)