Gauss-Jordan elimináció: pivotálás

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2021 november 24.

Lehetséges problémák: zérus elem

A Gauss-Jordan elimináció során többféle probléma is felléphet.

Lehetséges problémák: zérus elem

A Gauss-Jordan elimináció során többféle probléma is felléphet.

Probléma: 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 Gauss-Jordan elimináció során többféle probléma is felléphet.

Probléma: zérus elem a főátlóban

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

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

Megoldás: sorcsere

- az egyenletek sorrendje tetszőleges
- 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

$\Rightarrow$ *részleges pivotolás*.

Lehetséges problémák: nagyon kis elem

Probléma: 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 ?

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

Ezt nevezeik *implicit pivotálásnak*

Mindig pivotáljunk!

- meg lehet mutatni, hogy a kerekítési hibák miatt a Gauss-Jordan eljárás pivotálás nélkül általában numerikusan nem stabil
- tehát ha az A mátrixok elég nagy, akkor a kerekítési hibák miatt akkor is pontatlan lesz az eredmény, ha a fentebb említett problémák nem lépnek fel
- ezért a Gauss-Jordan eljárást mindig részleges és implicit pivotálással együtt használjuk!

Lehetséges problémák: szinguláris mátrix

- az oszlop főátló alatti része csupa 0

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

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.

Teljes pivotálás

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.

Teljes pivotálás

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.

Ezután mindkét oldalon az aktuális sorral lefelé eliminálunk.

Ezt nevezzük *teljes pivotolásnak*.

Teljes vs részleges pivotálás

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.

Tapasztalat: általában a részleges pivotálás “majdnem” olyan jó, mint a teljes

Az algoritmus építőkövei

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

- 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 algoritmus építőkövei

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

- 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)

Gondoljuk át a programfejlesztés lépéseit:

- az építőelemeket külön-külön megvalósítjuk és *teszteljük*!
- ezek után jöhet a fő algoritmus leprogramozása
- a programot többfajta mátrixon teszteljük (0 elem a főátlóban, stb)