Gauss – módszer

Hevesi Anikó tanárnő jegyzete

A **Gauss – módszer** az **egyenletrendszer** mátrixát ekvivalens átalakítások segítségével úgy rendezi át, hogy a mátrix **főátlójában egyesek, a diagonális alatt pedig nullákat kapjunk**. A diagonális felett bármilyen szám valós lehet.

$$\left.\begin{array}{ccc} 1 & . & . \\ 0 & 1 & . \\ 0 & 0 & 1 \end{array}\right| \begin{array}{c} x1 \\ x2 \\ x3 \end{array}$$

Az ekvivalens átalakítások alkalmazása jól látható egy példán. Az egyenletrendszerünk:

$$\begin{aligned} 2x_1 + 4x_2 &= 7 \\ 3x_1 + 6x_2 - x_3 &= 8 \\ x_1 + 5x_2 + x_3 &= 1 \end{aligned}$$

Az egyenletrendszer mátrixa a következő:

$$\left[\begin{array}{ccc|c} 2 & 4 & 0 & 7 \\ 3 & 6 & -1 & 8 \\ 1 & 5 & 1 & 1 \end{array}\right]$$

A **mátrix rendezését oszloponként végzem**. Elöször az első oszlopot hozom a kívánt alakra, majd a másodikat,…

Elöször a főátlón lévő elem helyére kell egyest tennem. **Ezt úgy érhetem el, hogy az első sort elosztom azzal a számmal, amely a főátlóban van**. Előfordulhat, hogy a főátlóban lévő elem nulla. Ebben az esetben **kicserélem a sort egy olyan alatta lévő sorral, ahol a megfelelő oszlopban található szám nem nulla**. (Ha nem találok ilyet, akkor az egyenletrendszernek nincs megoldása) Ezután elérem a már említett módon, hogy a főátlón egyeseket kapjak

$$\left[\begin{array}{ccc|c} 2 & 4 & 0 & 7 \\ 3 & 6 & -1 & 8 \\ 1 & 5 & 1 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 3 & 6 & -1 & 8 \\ 1 & 5 & 1 & 1 \end{array}\right]$$

Ezután az oszlop főátló alatii elemeit nullázom a következőképpen: minden sorból kivonom a diagonálison lelő elemet tartalmazó sor annyiszorosát, amilyen szám van azon a helyen, ahol nullát akarok kapni.

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 3 & 6 & -1 & 8 \\ 1 & 5 & 1 & 1 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 0 & -1 & -5/2 \\ 0 & 3 & 1 & -5/2 \end{array}\right]$$

Így a mátrix első oszlopának alakja már megfelelő. A második oszlopban ugyanígy járok el. Kezdem a főátlón levő elemmel. A mi mátrixunkban ez nulla, tehát kicserélem a második sort a harmadikkal.

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 0 & -1 & -5/2 \\ 0 & 3 & 1 & -5/2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 3 & 1 & -5/2 \\ 0 & 0 & -1 & -5/2 \end{array}\right]$$

Végig osztom a sort mátrix[1,1]-vel (második sor második oszlop csak 0-tól indexelve), így a diagonálison egyeseket kapok.

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 3 & 1 & -5/2 \\ 0 & 0 & -1 & -5/2 \end{array}\right] \sim \left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 1 & 1/3 & -5/6 \\ 0 & 0 & -1 & -5/2 \end{array}\right]$$

Így folytatom a mátrix rendezését oszloponként, amíg a mátrix kívánt alakú nem lesz:

$$\left[\begin{array}{ccc|c} 1 & 2 & 0 & 7/2 \\ 0 & 1 & 1/3 & -5/6 \\ 0 & 0 & 1 & 5/2 \end{array}\right]$$

A főátlóban egyeseket kaptam, a főátló alatt pedig nullákat. A mátrixhoz tartozó egyenletrenszer a következő:

$$\begin{aligned} x_1 + 2x_2 \qquad\quad &= 7/2 \\ x_2 - 1/3x_3 &= -5/6 \\ x_3 &= 5/2 \end{aligned}$$

A mátrix utolsó sorában mindenhol nulla van, csak a főátlóban van egyes**. Így az utolsó ismeretlen már ki van fejezve**. A többi ismeretlen alulról felfelé haladva fokozatos visszahelyettesítéssel tudom kiszámolni. Az egyenletrenszer gyökei a mátrix n-ik oszlopában lesznek.

$$\begin{aligned} & x_3 = \frac{5}{2} \\ & x_2 + \frac{1}{3} \times \frac{5}{2} = -\frac{5}{6} \\ & x_2 = -\frac{10}{6} \\ & x_1 = 2 \times \left(-\frac{10}{6}\right) + 0 \times \frac{5}{2} = \frac{7}{2} \\ & x_1 = \frac{41}{6} \end{aligned}$$

Az algoritmus menete:

1. Végig megyek a diagonálison
   a. Ha az adott cellában nulla van, keresek egy olyan sort alatta, amelyben az adott oszlopban valamilyen szám van (ha nincs, nincs megoldás)
   b. Elosztom a cella értékével az egész sort, így 1-est kapok a cellában.
   c. Oszlop alatti elemeket nullázom
2. Visszafejtem a megoldást az utolsó elemtől és megkapom a gyököket

```python
def oszt(sor,oszto,matrice):
    """ sor-adik sor osztasa egy adott szammal(oszto) """

    for k in range(len(matrice)+1):
        matrice[sor][k] = matrice[sor][k]/oszto


def csere(x,y,matrice):
    """ x-edik és az y-adik sor cseréje """
    temp = matrice[x]
    matrice[x] = matrice[y]
    matrice[y] = temp


def keres(sor,oszlop,matrice):
    """ Keres egy olyans sort, ahol az adott oszlopban nem 0 van és kicseréli azokat"""
    index = -1

    for k in range(sor+1,len(matrice)):
        if matrice[k][oszlop] != 0:
            index = k
            break

    if index == -1:
        print("Nincs megoldas")
    else:
        csere(sor,index,matrice)


def osszead(x,y,seged,matrice):
    """ az x-dik sor segedszeresenek hozzaadasa az y/ik sorhoz"""
    for k in range(len(matrice)+1):
        matrice[y][k] -= matrice[x][k]*seged



def gyokok(matrice):
    """ A megfelelőre alakra hozott mátrixból a gyökök meghatározása"""
    n = len(matrice)
    for i in range(n,-1,-1):
        for j in range(i+1,n):
            matrice[i][n] = matrice[i][n] - matrice[i][j]*matrice[j][n]




def gauss(matrice):
    """ Gauss módszer """
    for i in range(len(matrice)):
        if matrice[i][i] == 0:
            keres(i,i,matrice)

        oszt(i,matrice[i][i],matrice)

        for j in range(i+1,len(matrice)):

            osszead(i,j,matrice[j][i]/matrice[i][i],matrice)

    gyokok(matrice)
```