Alkalmazott matematika

Baran Ágnes

Lineáris egyenletrendszerek

Lineáris algebra Octave/Matlab-bal

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ha

$$A = \begin{bmatrix} -2 & -1 & 4 \\ 2 & 3 & -1 \\ -4 & -10 & -5 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 1 \\ -12 \end{bmatrix}$$

Megoldás. Használjuk a `backslash` operátort!

```
>>A=[-2 -1 4; 2 3 -1; -4 -10 -5];
>>b=[3; 1; -12];
>>x=A\b
x=
    3
   -1
    2
```

Ügyeljünk rá, hogy a b oszlopvektorként legyen megadva!

Ha az egyenletrendszer kibővített mátrixával meghívjuk az `rref` függvényt:

```
>>rref([A b])
ans=
   1 0 0  3
   0 1 0 -1
   0 0 1  2
```

akkor láthatjuk, hogy a Gauss-Jordan elimináció eredményeként valóban így állítható elő a b vektor az A oszlopvektoraiból, amelyek lineárisan függetlenek, tehát a megoldás egyértelmű.

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ha

$$A = \begin{bmatrix} -4 & -4 & 2 \\ -2 & -7 & 3 \\ 2 & 12 & -5 \end{bmatrix}, \quad b = \begin{bmatrix} -2 \\ 6 \\ -13 \end{bmatrix}$$

Megoldás. Próbálkozzunk ismét a `backslash` operátorral!

```
>>A=[-4 -4 2; -2 -7 3; 2 12 -5];
>>b=[-2; 6; -13];
>>x=A\b
Warning: Matrix is singular to working precision
x=
   NaN
   NaN
   NaN
```

A Matlab arra figyelmeztetett, hogy a mátrix szinguláris (valóban, $\det(A) = 0$).

Próbálkozzunk az `rref` függvénnyel!

```
>>rref([A b])
ans=
   1.0000       0 -0.1000  1.9000
        0  1.0000 -0.4000 -1.4000
        0       0       0       0
```

Azt látjuk, hogy a mátrix oszlopvektorai lineárisan függőek, de a b vektor benne van az oszlopvektorok által felfeszített térben. Tudjuk, hogy ilyenkor az egyenletrendszernek végtelen sok megoldása van, ezek közül egy:

$$x = \begin{bmatrix} 1.9 \\ -1.4 \\ 0 \end{bmatrix}$$

Ha az egyenletrenszer összes megoldását szeretnénk tudni, akkor használjuk a `null` függvényt, amely előállítja a nulltér egy bázisát:

```
>>p=null(A,'r')
p=
   1/10
   2/5
     1
```

(az 'r' opció hatására a vektor racionális alakban jelenik meg)
Ezek szerint a lineáris egyenletrendszer általános megoldása:

$$\begin{bmatrix} 1.9 \\ -1.4 \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} 1/10 \\ 2/5 \\ 1 \end{bmatrix}$$

ahol $\lambda \in \mathbb{R}$.

Példa

Oldjuk meg az $Ax = b$ lineáris egyenletrendszert, ahol

$$A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 6 \\ 9 \\ 12 \end{bmatrix}$$

Megoldás. A `backslash` operátorral azt kapjuk, hogy

```
>>x=A\b
x=
   1.0000
   2.7000
```

Könnyen látható, hogy ez nem megoldása az egyenletrendszernek.

Az `rref` függvénnyel:

```
>>rref([A b])
ans=
   1 0 0
   0 1 0
   0 0 1
   0 0 0
```

láthatjuk, hogy az alapmátrix rangja 2, a kibővített mátrixé 3, az egyenletrendszer ellentmondásos.

Ellentmondásos lineáris egyenletrendszerek esetén a backslash operátor egy olyan x vektort ad vissza, melyre az Ax és b vektorok eltérése euklideszi normában a legkisebb (azaz $\|Ax - b\|_2$ minimális). Ilyenkor azt mondjuk, hogy x az egyenletrendszer legkisebb négyzetes értelemben vett megoldása.

1. feladat

Oldja meg Octave/Matlab-bal az $Ax = b$ lineáris egyenletrendszert, ahol

(a)

$$A = \begin{bmatrix} 2 & -3 & 1 & 1 \\ -1 & 3 & 4 & 7 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

(b)

$$A = \begin{bmatrix} 2 & 1 \\ -3 & 4 \\ 5 & -1 \end{bmatrix}, \quad b = \begin{bmatrix} -5 \\ 24 \\ -23 \end{bmatrix}$$

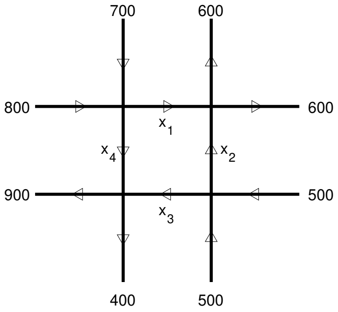
(c)

$$A = \begin{bmatrix} 2 & 1 & 5 & 0 \\ -3 & 4 & -13 & 22 \\ 5 & -1 & 16 & -14 \\ 1 & 1 & 2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 12 \\ 81 \\ -33 \\ 15 \end{bmatrix}$$

Hasznos: ha az x racionális elemű vektor koordinátáit nem tizedestört alakban akarjuk látni, akkor használhatjuk a `rats(x)` utasítást, vagy a kiiratás formátumát állítsuk át: `format rat`

2. feladat

A következő feladat megoldásához használja a Matlab-ot.
Az alábbi ábrán négy egyirányú útszakasz látható, ahol a forgalom a nyilaknak megfelelő irányban halad. Az egyes ágakba óránként be- és kilépő járművek számát az utak végén jelöltük. Adja meg az x_1-gyel jelölt helyen óránként áthaladó járművek minimális és maximális számát. Mennyi jármű halad át ekkor a többi jelölt helyen?



3. feladat

Olvassa el a `rank` függvény help-jét!

Írjon egy Matlab függvényt, mely adott A mátrix és b vektor esetén megadja, hogy az $Ax = b$ lineáris egyenletrendszer egyértelműen megoldható, ellentmondásos, vagy végtelen sok megoldása van.

Egészítse ki a kódot úgy, hogy ha az A mátrix és a b vektor mérete nem kompatibilis, akkor erre figyelmeztesse a felhasználót!

4. feladat

Matlab-ban egy konkrét A mátrix és b oszlopvektor megadása után kiadtuk az

```
>> x=A\b;
```

parancsot, amely után az `x` változó értéke a következő lett:

```
x=
    -1
     0
     1
     2
```

Mit mondhatunk az alábbi állításokról (külön-külön)? (Igaz/Hamis/Nem eldönthető). Válaszait indokolja.

(a) A b vektornak 4 eleme van.

(b) Az A mátrixnak 4 sora van.

(c) Az A mátrixnak 4 oszlopa van.

(d) $Ax = b$

(e) $x = A^{-1}b$

5. feladat

Egy hajó folyásirányban $t_1 = 4$ órán át halad egy folyón, így $s_1 = 140$ km-t tesz meg. Ott megfordul, és $t_2 = 3$ órán át halad folyásiránnyal szemben, ekkor $s_2 = 75$ km-t tesz meg. Ha tudjuk, hogy a hajó parthoz viszonyított sebessége végig állandó volt, akkor mekkora ez a sebesség és milyen sebességgel folyik a folyó? Írjon egy Matlab függvényt, mely tetszőleges t_1, t_2, s_1, s_2 értékek esetén a keresett mennyiségekkel tér vissza (vagy egy megfelelő hibaüzenettel).

Több jobboldali vektor

Példa

Oldjuk meg az $Ax = b$ és $Ax = c$ egyenletrendszereket, ha

$$A = \begin{bmatrix} -2 & -1 & 4 \\ 2 & 3 & -1 \\ -4 & -10 & -5 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ 1 \\ -12 \end{bmatrix}, \quad c = \begin{bmatrix} 17 \\ 1 \\ -42 \end{bmatrix}$$

Megoldás. Mivel a két rendszer mátrixa azonos, ezért megoldhatjuk őket egyszerre.

```
>>A=[-2 -1 4; 2 3 -1; -4 -10 -5];
>>b=[3; 1; -12]; c=[17; 1; -42];
>>x=A\[b c]
x=
    3  -2
   -1   3
    2   4
```

Több jobboldali vektor

Nagyméretű mátrixok esetén a futási időt jelentősen befolyásolhatja, hogy az azonos mátrixszal adott rendszereket egyszerre, vagy külön-külön oldjuk meg:

```
>> A=rand(10000);
>> b=ones(10000,1);
>> c=zeros(10000,1);
>> tic;x=A\[b,c];toc
Elapsed time is 6.116513 seconds.
>> tic;x=A\b; x2=A\c; toc
Elapsed time is 11.571959 seconds.
```

(A fenti eredmény egy Intel Core i5-4590 processzorral, 7.7 GiB memóriával rendelkező gépen született).

LU-felbontás

`[L,U,P]=lu(A)`

Ekkor

- L: alsóháromszög mátrix, átlójában csupa 1-es
- U: felsőháromszög mátrix
- P: permutációs mátrix

úgy, hogy $PA = LU$.

`[L1,U1]=lu(A)`

Ekkor $A = L1 \cdot U1$ úgy, hogy $U1$ megegyezik az előző U mátrixszal és $L1 = P^T L$.

Több jobboldali vektor

Ha több lineáris egyenletrendszert kell megoldanunk, ahol a mátrix azonos, a jobboldali vektorok különbözőek, de a jobboldali vektorok nem állnak egyszerre rendelkezésre, akkor a következő utasításokat használjuk:

Egyetlen egyszer, a rendszerek megoldása előtt készítsük el a mátrix LU-felbontását:

```
>> [L,U]=lu(A);
```

Ahányszor egy újabb *b* jobboldali vektor rendelkezésünkre áll, adjuk ki az

```
>> x=U\(L\b);
```

utasítást, amivel megkapjuk az adott jobboldali vektor esetén a rendszer megoldását.

Cholesky-felbontás

- `chol(A)`
  elkészítí az A mátrix Cholesky-felbontását, a felbontásban szereplő felsőháromszög mátrixszal tér vissza.
- `chol(A,'lower')`
  elkészítí az A mátrix Cholesky-felbontását, a felbontásban szereplő alsóháromszög mátrixszal tér vissza.

Ha az A nem pozitív definit, akkor nem létezik a felbontás, hibaüzenetet kapunk.
A felbontás létezéséhez a mátrixnak szimmetrikusnak kell lennie, ezt nem vizsgálja, de első esetben csak az A felső-, a másodikban az alsóháromszög részét használja.

6. feladat

Tekintse az $Ax = b$ lineáris egyenletrendszert, ahol

$$A = \begin{bmatrix} 10^{-17} & 1 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Ennek megoldásához először alkalmazza a

```
c=A(2,2)-A(1,2)*A(2,1)/A(1,1);
d=b(2)-b(1)*A(2,1)/A(1,1);
x(2)=d/c
x(1)=(b(1)-A(1,2)*x(2))/A(1,1)
```

utasításokat (amik a sorcsere nélküli Gauss-eliminációnak felelnek meg a 2×2-es $Ax = b$ lineáris egyenletrendszer esetén), majd ezek után a $x = A\backslash b$ utasítást. Magyarázza meg a tapasztalt jelenséget!

7. feladat

Legyen `A=pascal(10)` (azaz A a 10×10-es Pascal mátrix, ami egy szimmetrikus, pozitív definit mátrix), `x=ones(10,1)` és definiálja a b vektort úgy, hogy `b=A*x`. Oldja meg az $Ax = b$ rendszert az `lu`, `chol` és $A \backslash b$ utasításokat alkalmazva (használjon „format long"-ot)!

Mátrix inverze Octave/Matlab-bal

Az `inv` függvénnyel számítható. Ha a mátrix nem négyzetes, vagy a determinánsa 0 (vagy 0-hoz közeli), akkor hibaüzenetet, illetve figyelmeztetést kapunk.

Nagyméretű mátrixok inverzének kiszámítása túl költséges lehet. Csak akkor számoljuk ki, ha ténylegesen szükségünk van az inverzre.

Pl. az $Ax = b$ négyzetes mátrixú lineáris egyenletrendszer megoldása $x = A^{-1}b$ módon kb háromszor annyi műveletbe kerül, mint az $x = A\backslash b$ megoldás.

Ritka mátrixok

A `sparse` függvény

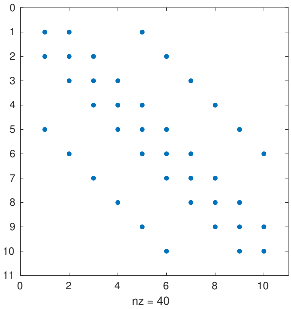
```
>> A=[-1.1 0 0 2; 0 0 2 0; 0 -1 0 1;0 0 0 3]
A =
   -1.1000         0         0    2.0000
         0         0    2.0000         0
         0   -1.0000         0    1.0000
         0         0         0    3.0000

>> S=sparse(A)
S =
   (1,1)      -1.1000
   (3,2)      -1.0000
   (2,3)       2.0000
   (1,4)       2.0000
   (3,4)       1.0000
   (4,4)       3.0000
```

Definiáljunk egy olyan 10×10-es ritka mátrixot, melyben csak 5 átlóban vannak 0-tól különböző elemek:

```
>> d=ones(10,1);
>> S=spdiags([d d -4*d d d],[-4 -1 0 1 4],10,10);
```

Megnézhetjük a nemnulla elemek elhelyezkedését:

```
>> spy(S)
```



A nemnulla elemek száma: `nnz(S)`

Hasonlítsuk össze egy nagyméretű ritka mátrix esetén a tárigényt a különböző tárolási módok esetén:

```
>> d=ones(10000,1);
>> S=spdiags([d d -4*d d d],[-4000 -1 0 1 4000],10000,10000);
>> F=full(S);
>> whos S F
```

Vizsgáljuk meg egy mátrix-vektor szorzás futási idejét:

```
>> x=rand(10000,1);
>> tic;b=S*x;toc
>> tic;b=F*x;toc
```

és egy lineáris egyenletrendszer megoldásának futási idejét

```
>> tic;y=F\b;toc
>> tic;y=S\b;toc
```