

Matematikai programok

Mátrixalapú nyelvek – octave

Wetl Ferenc
Algebra Tanszék



1 Gyors áttekintés

2 Programozás







3 Vektorosítás

1 Gyors áttekintés

2 Programozás

3 Vektorosítás

Mátrixalapú nyelvek

-  **MATLAB**® (**matrix laboratory** – The Language of Technical Computing)
 - kereskedelmi program
 - többmillió felhasználó
 - numerikus matematika, jel- és képfeldolgozás, kommunikáció, irányítástechnika, pénzügyi matematika, . . .
 - <http://www.mathworks.com>
- szabad klónok:
 -  **octave**: GNU GPL, honlap www.gnu.org/software/octave/
 -  **Scilab**: CeCILL license (GPL compatible), honlap www.scilab.org
(fő fejlesztő: [INRIA](http://www.inria.fr), Franciaország)
 -  **FreeMat**: GPL, honlap freemat.sourceforge.net/
- Hasonló szoftverek:
 -  **julia** **julia**: MIT licensed, (high-performance dynamic programming language for technical computing), honlap julialang.org/
 -  **R**: GPL, statisztikai számításokhoz

Számolás

- valós és komplex számok

- `octave> 1 + 1`
`ans = 2`

```
octave> 2^23  
ans = 8388608
```

```
octave> 2^-3  
ans = 0.12500
```

```
octave> 2^123  
ans = 1.0634e+37
```

```
octave> (1 + 2i) / (3 - 1i)  
ans = 0.10000 + 0.70000i
```

```
octave> (1 - 1i)^8  
ans = 16
```

Számolás mátrixokkal

- valós és komplex mátrixok

```

octave> [1 1; 3 3] + [1 2; 2 3]*[3 1; 2 1]^-1
ans =
   -2.00000    6.00000
   -1.00000   10.00000
  
```

```

octave> [1 1; 3 3] + [1 2; 2 3] * [2 1; 1 1]^-1
ans =
   3.3307e-16    4.0000e+00
   2.0000e+00    7.0000e+00
  
```

```

octave> [1; 2]
ans =
    1
    2
  
```

```

octave> ans'
ans =
    1    2
  
```

```
■ octave> zeros (2, 3)
ans =
    0    0    0
    0    0    0

octave> ones (1, 4)
ans =
    1    1    1    1

octave> eye (3)
ans =
Diagonal Matrix
    1    0    0
    0    1    0
    0    0    1

octave> diag ([2 3 1])
ans =
Diagonal Matrix
    2    0    0
    0    3    0
    0    0    1
```

Mátrixok osztása és inverze

- $\mathbf{A}/\mathbf{B} = \mathbf{C}$ jelentése: $\mathbf{A} = \mathbf{CB}$
- $\mathbf{A}\backslash\mathbf{B} = \mathbf{C}$ jelentése: $\mathbf{B} = \mathbf{AC}$
- Az $\mathbf{Ax} = \mathbf{b}$ egyenletrendszer megoldására két lehetőség:
(1) $\mathbf{x} = \mathbf{A}\backslash\mathbf{b}$, (2) invertálható együtthatómátrix esetén $\mathbf{A}^{-1}\mathbf{b}$
- Oldjuk meg a $\left\{ \begin{array}{l} x - 3y = 15 \\ 4x + 2y = 18 \end{array} \right\}$ egyenletrendszert kétféleképp!

```
octave> [1 -3; 4 2] \ [15; 18]
ans =
     6
    -3
```

```
octave> [1 -3; 4 2]^(-1) * [15; 18]
ans =
     6
    -3
```


- Mátrixok osztása akkor is elvégezhető, ha a „nevezőben” lévő mátrix nem invertálható (ekkor a később tanulandó általánosított inverzzel számol).
- Ezzel mindig megkapható az egyenletrendszer sortérbe eső megoldása, a redukált lépcsős alakkal az összes. Az egyenletrendszer legyen

$$x + y + 2z = 2$$

$$2x + 2y + 3z = 4$$

```
octave> [1 1 2; 2 2 3] \ [4; 8]
```

```
ans =
```

```
2.0000e+00
```

```
2.0000e+00
```

```
7.1054e-15
```

```
octave> rref ([1 1 2; 2 2 3])
```

```
ans =
```

```
1    1    0
```

```
0    0    1
```

Egészek

- A legtöbb számolás **duplapontosságú lebegőpontos** számokkal történik. Az egész típus inkább csak adatok tárolására vagy megjelenítésére való.
- **integer** b biten, vagy **unsigned integer**: int8, uint8, int16, uint16, int32, uint32, int64, uint64.

```
octave> 10 * rand (2, 3)
ans =
```

```
8.390913    0.500552    6.421794
9.608188    1.873848    0.028212
```

```
octave> int8 (ans)
ans =
```

```
8    1    6
10   2    0
```

Tartományok (ranges)

■ kezdet:lépésköz:vég

■ `octave> 1:4`

`ans =`

1 2 3 4

`octave> 4:1`

`ans = [] (1x0)`

`octave> 9:-3:1`

`ans =`

9 6 3

`octave> 1.1:.237:2.1`

`ans =`

1.1000 1.3370 1.5740 1.8110 2.0480

Változók

```
■ octave> a = 3
```

```
a = 3
```

```
octave> m = [
```

```
  1 2 a
```

```
  2 a 4]
```

```
m =
```

```
  1  2  3
```

```
  2  3  4
```

```
octave> m' * m
```

```
ans =
```

```
  5  8  11
```

```
  8  13  18
```

```
 11  18  25
```

Adattípusok: typeinfo, scalar, matrix, range, string

- valós és komplex skalár és mátrix, tartomány (range), karakterlánc...

```
octave> typeinfo (a)
```

```
ans = scalar
```

```
octave> typeinfo (1.23)
```

```
ans = scalar
```

```
octave> typeinfo (m)
```

```
ans = matrix
```

```
octave> typeinfo ([1 + 2i 1 - 2i])
```

```
ans = complex matrix
```

```
octave> typeinfo (1:4)
```

```
ans = range
```

```
octave> typeinfo("valami")
```

```
ans = string
```

Indexek

```
■ octave> M
```

```
M =
```

```
      8      1      6
     10      2      0
```

```
octave> M (1, 3)
```

```
ans = 6
```

```
octave> M (1, [2, 3, 1])
```

```
ans =
```

```
      1      6      8
```

```
octave> M (:, [2, 3, 1])
```

```
ans =
```

```
      1      6      8
      2      0     10
```

1 Gyors áttekintés

2 Programozás

3 Vektorosítás

Függvények

```
■ octave> function fv1  
    1 + 1  
endfunction
```

```
octave> fv1  
ans =  2
```

```
■ octave> function fv2 (a, b)  
    a^2 + b^2  
endfunction
```

```
octave> fv2 (3, 4)  
ans =  25
```



```

■ function [amegb, aszorb] = muvelet (a, b)
    amegb = a + b;
    aszorb = a * b;
endfunction
octave> [c d] = muvelet (2, 3)
c = 5
d = 6

```

■ A függvénydefiníció általános alakja (a színes rész opcionális):
 function *visszaadott érték = függvénynév (argumentumlista)*
 octave utasítások
 endfunction

Logikai értékek (1 = igaz, 0 = hamis, típusa bool)

```
■ octave> 4 > 1
ans = 1

octave> 4 < 1
ans = 0

octave> 4 == 1
ans = 0

octave> 4 >= 1
ans = 1

octave> (4 >= 1) == 1
ans = 1
octave> typeinfo (ans)
ans = bool

octave> a = true
a = 1
```

Feltételes utasítás

- `if` (*feltétel*)

octave utasítások

`else`

ez a rész

octave utasítások

kimaradhat

`endif`

Feltételes utasítás (példa)

```
■ function valosvagykomplex (a, b, c)
    d = b^2 - 4*a*c; # ; hogya ne irja ki d erteket
    if (d >= 0)
        "valos"
    else
        "komplex"
    endif
endfunction
```

```
octave> valosvagykomplex (1, 1, 1)
ans = komplex
```

```
octave> valosvagykomplex (1,2,1)
ans = valos
```

1 Gyors áttekintés

2 Programozás

3 Vektorosítás

Vektorosítás

- A cél, hogy a kódismétlés és a ciklusutasítások helyett vektor-/mátrixműveleteket használjunk.
- Így hatékonyabb, tömörebb kód jön létre.
- Írjunk kódot, mely egy adott (v_1, v_2, \dots, v_n) vektorból a különbségek $(v_2 - v_1, v_3 - v_2, \dots, v_n - v_{n-1})$ vektorát képzeli!

```
octave> l = [3 4 6 2 5 1]
```

```
l =
```

```
    3    4    6    2    5    1
```

```
octave> l(2:6) - l(1:5)
```

```
ans =
```

```
    1    2   -4    3   -4
```

- Alkalmazzuk az $f(x) = x^2 + 3x + 1$ képletet egy **mátrixra!**

```
octave> function keplet(a)
    a^2 + 3*a + 1
endfunction
octave> keplet( [1 1;0 2])
ans =
     5     7
     1    11
```

- Alkalmazzuk az $f(x) = x^2 + 3x + 1$ képletet egy **mátrix minden elemére!**

```
octave> function keplet(a)
    a .^ 2 + 3 .* a + 1
endfunction
octave> keplet ([1:3 5])
ans =
     5    11    19    41
```

Kérdések

- Mire és hogyan használhatók az `eye`, `ones`, `zeros` és `diag` parancsok?
- Hozzunk létre egy `m` nevű véletlen 4×5 -ös mátrixot, melynek elemei 0-tól 9-ig terjedő egészek!
- Soroljuk fel a tanult adattípusokat! Melyik paranccsal jutunk ehhez az információhoz egy adatról?
- Írjunk függvényt, melynek x a bemenete és az $x + \frac{4}{x^2} + \frac{1}{x^3}$ értéket adja vissza!
- Írjunk meg ugyanezt a függvényt, de úgy, hogy egy mátrixra alkalmazva, annak minden elemére kiszámítsa az értékét!
- Mi az eredménye a `1 == (2 > 4)` és a `0 == (2 > 4)` műveletnek? Ezeknek mi az adattípusa?
- Egy 4×5 -ös véletlen mátrixon végezzünk elemi sorműveleteket!
- Írjunk függvényt, mely az argumentumába írt számról eldönti az előjelét és azt karakterláncként kiírja!

Kérdések

- Képezzünk egy 10×10 -es mátrixot, melynek blokkmátrix alakja:

$$\begin{bmatrix} \mathbf{O}_5 & \mathbf{I}_5 \\ \mathbf{I}_5 & \mathbf{O}_5 \end{bmatrix}$$

- Oldjuk meg az

$$x - 2y = 1$$

$$2x + y = 7$$

egyenletrendszert mátrixosztással, és az inverzzel való szorzással.

- Számítsuk ki az

$$x - 2y + 3z = -1$$

$$2x + y + z = 3$$

egyenletrendszer egyetlen sortérbe eső megoldását, majd az együtthatómátrix redukált lépcsős alakját!