

4. Matlab - matice, definovanie a adresovanie

Foltin Martin · MATLAB/Comsol

15.05.2009



Matica je základným dátovým objektom v Matlabe. Veď aj samotný názov Matlab v sebe skrýva maticu (MATrix LABoratory). V tejto časti seriálu si ukážeme ako maticu definovať ako v nej vyhľadávať prvky a submatice no a v závere si vysvetlíme niektoré maticové operácie.

V lineárnej algebre sa stretávame so základným dátovým usporiadaním ako je skalár, vektor, alebo matica. Matlab všetky tieto objekty vníma ako jeden dátový objekt - matice. Skalár je špecifický prípad matice s rozmerom 1×1 , vektor je zas maticou $1 \times N$ alebo $N \times 1$ no a matica má rozmer $M \times N$. Matice môžeme vkladať do premenných pomocou priradenia. Ani v tomto prípade nemusíme premenné špeciálne deklarovať. Ich deklarácia sa udeje v momente prvého použitia. Skalár do premennej priradíme pomocou symbolu priradenia =

```
>> a=4
a =
    4
```

V premennej a sa teraz nachádza matica s rozmerom 1×1 a má hodnotu 4. Vektor priradíme tiež pomocou symbolu =. Ak definujeme riadkový vektor, prvky sú oddelené medzerou alebo čiarkou. V prípade stĺpcového vektora oddeľujeme prvky bodkočiarkou. Všetky prvky sú uzavreté v hranatých zátvorkách.

```
>> v=[10 20 30 40]
v =
    10 20 30 40
```

V premennej v je teraz uložená matica s rozmerom 1×4 (prvá hodnota vypovedá o počte riadkov, druhá o počte stĺpcov) a hodnotami 10, 20, 30 a 40. V prípade, že potrebujeme pracovať so stĺpcovým vektorom máme tieto možnosti jeho definície.

- definovať riadkový vektor a transponovať ho symbolom apostrof [10 20 30 40]'
- oddeliť prvky bodkočiarkou [10 ; 20 ; 30 ; 40]

Oba zápisy sú ekvivalentné.

Inou možnosťou ako vytvoriť vektor, je využiť vstavané funkcie. Vysvetlime si niektoré. Často používaný spôsob tvorby vektora je pomocou dvojbodky. Vektor sa defunuje okrajovými hodnotami a krokom s akým majú byť prvky od seba vzdialené.

od:krok:do

```
>> v=0:0.2:1
```

v =

```
0 0.2000 0.4000 0.6000 0.8000 1.0000
```

Iný spôsob definovania je pomocou príkazu **linspace**. Argumenty tejto funkcie sú ľavý okraj intervalu, pravý okraj intervalu a počet prvkov výsledného vektora. Jednotlivé prvky budú od seba rovnako vzdialené. Jedná sa o lineárne delenie intervalu.

```
>> v=linspace(1,8,3)
```

v =

```
1.0000 4.5000 8.0000
```

Obdobne pracuje aj príkaz **logspace**. Rozdiel je v tom, že delenie je logaritmické. Parametre tohto príkazu sú 3. `logspace(x1,x2,N)` a znamenajú definovanie okrajov a počet delení. Ľavý okraj je 10^{x1} , pravý 10^{x2} a počet delení je N.

```
>> v=logspace(1,3,3)
```

v =

```
10 100 1000
```

Maticu môžeme definovať tiež niekoľkými spôsobmi. Často používaným je vymenovanie prvkov. Jednotlivé prvky sú od seba oddelene medzerou, riadky sú oddelené bodkočiarkou. Všetky prvky sú uzavreté v hranatých zátvorkách.

```
>> A=[1 2 3;4 5 6;7 8 9]
```

A =

```
1 2 3
4 5 6
7 8 9
```

Matlab obsahuje niekoľko vstavaných funkcií, ktoré generujú špeciálne matice.

ones/zeros

Funkcia `ones` vracia maticu s rozmerom $M \times N$, kde sú všetky prvky 1. Funkcia `zeros` pracuje rovnako s tým rozdielom, že matica je plná núl.

```
>> A=ones(3,4)
```

A =

```
1 1 1 1
1 1 1 1
1 1 1 1
```

eye

Funkcia `eye` vracia maticu ktorá ma na diagonále jednotky. Matica nemusí byť len

štvorcová.

```
>> eye(3)
ans =
    1 0 0
    0 1 0
    0 0 1
```

diag

Funkciou diag vytvoríme maticu, ktorá bude mať diagonálne prvky také, aký je parameter tejto funkcie. Pozíciu prvkov je možné posúvať nad (kladný parameter) alebo pod (záporný parameter) hlavnú diagonálu druhým parametrom.

```
>> A=diag([1 2 3])
A =
    1 0 0
    0 2 0
    0 0 3
```

```
>> A=diag([1 2 3],1)
A =
    0 1 0 0
    0 0 2 0
    0 0 0 3
    0 0 0 0
```

rand

Táto funkcia naplní maticu náhodne zvolenými číslami. Čísla sú z intervalu $<0, 1>$ s rovnomerným rozdelením.

```
>> A=rand(3,5)
A =
    0.8147    0.9134    0.2785    0.9649    0.9572
    0.9058    0.6324    0.5469    0.1576    0.4854
    0.1270    0.0975    0.9575    0.9706    0.8003
```

randn

Funkcia randn plní tiež maticu náhodnými číslami. Rozdiel je v tom, že čísla sú určované podľa normálneho rozdelenie. Stredná hodnota je teda 0.

```
>> A=randn(2,4)
A =
   -0.2050    1.4897    1.4172   -1.2075
   -0.1241    1.4090    0.6715    0.7172
```

Adresovanie matíc

Matica pozostáva z prvkov usporiadaných do riadkov a stĺpcov. Každý prvok má preto jasne definovanú pozíciu. V Matlabe je vždy pri adresovaní prvý parameter riadok a druhý stĺpec. Definujme si maticu A.

```
>> A=[1 2 3 4; 5 6 7 8;9 10 11 12; 13 14 15 16]
```

```
A =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

Prvok 10 má súradnice (3,2). Je teda na treťom riadku a v druhom stĺpci. Ak ho potrebujeme z matice A vybrať a priradiť do inej premennej použijeme takýto zápis:

```
>> c=A(3,2)
```

```
c =
    10
```

Ak by sme chceli zmeniť prvok na pozícii prvku 10, tak postupujeme obdobne.

```
>> A(3,2)=333
```

```
A =
     1     2     3     4
     5     6     7     8
     9    333    11    12
    13    14    15    16
```

Pre prípad, že potrebujeme pracovať s riadkom alebo stĺpcom matice, využijeme operátor dvojbodka (:). Tento operátor symbolizuje výber všetkých prvkov. Zápis A(:,3) teda vyberie z matice A všetky riadky (dvojbodka je na prvej pozícii) a 3. stĺpec.

```
>> A(:,3)
```

```
ans =
     3
     7
    11
    15
```

Obdobne môžeme postupovať pri výbere riadku.

```
>> A(2,:)
```

```
ans =
     5     6     7     8
```

Operátor : môžeme využiť aj pri vyberaní submatice z matice.

```
>> A(2:3,2:4)
```

```
ans =
```

```

6    7    8
333 11   12

```

Uvedený zápis znamená, že z matice A vyberáme prvky na druhom a treťom riadku (2:3) a v druhom, treťom a štvrtom stĺpci (2:4).

Ďalším operátorom, ktorý môžeme využiť pri adresovaní prvkov matice, je operátor `end`. Tento operátor adresuje poslednú pozíciu riadku alebo stĺpca v závislosti od toho na ktorom mieste ho použijeme. Pozícia `(end,end)` adresuje prvok v poslednom riadku a poslednom stĺpci.

```

>> A(end,end)
ans =
    16

```

Operátor `end` môžeme využiť aj pri tvorbe submatice

```

>> A(3:end,2:end)
ans =
    333    11    12
    14    15    16

```

Adresovanie vektorov pracuje na rovnakých princípoch. Rozdiel je len v tom, že používame len jeden parameter pre určenie pozície. To či je vektor riadkový alebo stĺpcový nehrá rolu.

Rozmer matíc

Na určenie rozmeru matíc používame dve funkcie. Jedná sa o príkazy `size` a `length`. Príkaz `length` by sa mal používať pri určovaní veľkosti vektora, ale dá sa použiť aj pri práci s maticami. Tento príkaz vracia len jeden parameter. Ak ako argument zadáme vektor, tak vráti počet prvkov tohto vektora. Ak by sme ale ako argument použili maticu, tak vráti počet riadkov, alebo stĺpcov v závislosti od toho čoho je viac. Definujme si ľubovoľnú maticu s rozmerom 500×2 a použime ju ako argument pre funkciu `length`.

```

>> A=rand(500,2);
>> length(A)
ans =
    500

```

Obdobne môžeme postupovať ak ako argument použijeme ľubovoľný vektor.

```

>> v=0:0.01:2;
>> length(v)
ans =
    201

```

Príkaz `size` je už určený špeciálne pre stanovenie rozmeru matice. Samozrejme, že ako argument môžeme použiť aj vektor. Funkcia `size` vracia dve hodnoty. Prvá ja počet riadkov argumentu a druhý počet stĺpcov. Výstupom je teda vektor ktorý ma dva stĺpce

a jeden riadok.

```
>> size(A)
ans =
    500 2
```

```
>> size(v)
ans =
     1 201
```

V tejto časti sme si vysvetlili ako definovať vektor a maticu. Jedná sa o elementárnu prácu s týmto dátovým typom. Podrobnejšie informácie k tejto téme nájdete v dokumentácii k Matlabu. V budúcej časti sa zameriame na matematické operácie definované pre prácu s maticami.
