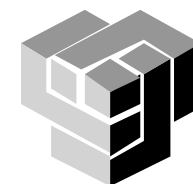


ИНТЕРАКТИВНА СИСТЕМА ЗА АНАЛИЗ, МОДЕЛИРАНЕ И
СИМУЛАЦИЯ (МАТЛАВ)

dimitrova@tu-sofia.bg
pct.tu-sofia.bg/dd/pik3



ОСНОВНИ ОБЕКТИ



Матрици

Определение

Матрица - правоъгълен масив от числа с размер $m \times n$

Вектор - матрица с един ред или един стълб, $1 \times n$, $m \times 1$

Скалар - матрица с размер 1×1

Въвеждане на матрици

чрез явен списък от елементи

като резултат от операции

Обръщение към матрица – чрез нейното име

Примери

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

```
A =
```

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

```
A =
```

$$\begin{bmatrix} 16 & 3 & 2 & 13 \\ 5 & 10 & 11 & 8 \\ 9 & 6 & 7 & 12 \\ 4 & 15 & 14 & 1 \end{bmatrix}$$


Вектори

- Списък от стойности, наречени елементи или компоненти
- Разделители: , ; интервал
- Ограничители: []
- Дължина на вектор: броят на елементите, length
- Вектор-ред и вектор-стълб

Примери:

Вектор - ред

```
>> v = [ 1 3, sqrt(5)]
v =
1.0000 3.0000 2.2361
>> length(v)
ans =
3
>> v2 = [3+ 4 5] % интервалът има значение
v2 =
7 5
>> v3 = [3 +4 5]
v3 =
3 4 5
```

Вектор - стълб

```
>> c = [ 1; 3; sqrt(5)]
c =
1.0000
3.0000
2.2361
>> c2 = [3
4
5]
c2 =
3
4
5
```



Операции с вектори

● Аритметични операции

- вектор и скалар
- вектори с еднаква дължина
- поелементни операции

● Примери:

```
>> v4 = 3*v
```

```
v4 =
```

```
3.0000 9.0000 6.7082
```

```
>> v + v3
```

```
ans =
```

```
4.0000 7.0000 7.2361
```

```
>> v5 = 2*v - 3*v3
```

```
v5 =
```

```
-7.0000 -6.0000 -10.5279
```

```
>> v + v2
```

```
??? Error using ==> +
```

```
Matrix dimensions must agree.
```



Генериране на вектори

Оператор :

равноотдалечени
елементи

дефинира се
интервал и стъпка

$a:b$ – резултатът е
вектор, съдържащ
числата от a до b
със стъпка 1

$a:c:b$ – резултатът е
вектор, съдържащ
числата от a до b
със стъпка c

```
>> v1 = 1:7
```

```
v1 =
```

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

```
>> v2 = 1:0.25:2
```

```
v2 =
```

```
1.0000 1.2500 1.5000 1.7500 2.0000
```



Генериране на вектори

● В резултат от операция

```
>> w = [1 2 3], z = [8 9]
```

```
>> cd = [2*z, -w]
```

```
w =
```

```
1 2 3
```

```
z =
```

```
8 9
```

```
cd =
```

```
16 18 -1 -2 -3
```



Други операции с вектори

Задаване на стойност на елемент

```
>> vec=[6 8 2 4];
```

```
>> vec(2)=3
```

```
vec =
```

```
6 3 2 4
```

Сортиране на елементите

```
>> sort(vec)
```

```
ans =
```

```
2 3 4 6
```

Извличане на стойности на елементи

```
>> vec(4)
```

```
ans =
```

```
4
```

```
>> vec(2:3)
```

```
ans =
```

```
3 2
```



Други операции с вектори

```
vec =
```

```
6 3 2 4
```

```
>> min(vec)
```

```
ans =
```

```
2
```

```
>> max(vec)
```

```
ans =
```

```
6
```

```
>> sum(vec)
```

```
ans =
```

```
15
```

Описателна статистика

```
>> mean(vec)
```

```
ans =
```

```
3.7500
```

```
>> std(vec)
```

```
ans =
```

```
1.7078
```

```
>> cumsum(vec)
```

```
ans =
```

```
6 9 11 15
```

```
>> diff(vec)
```

```
ans =
```

```
-3 -1 2
```



Транспониране на вектори: '

```
>> w, w', c, c'
```

```
w =
```

```
1 -2 3
```

```
ans =
```

```
1
```

```
-2
```

```
3
```

```
c =
```

```
1.0000
```

```
3.0000
```

```
2.2361
```

```
ans =
```

```
1.0000 3.0000 2.2361
```

```
>> t = w + 2*c'
```

```
t =
```

```
3.0000 4.0000 7.4721
```



Транспониране на вектори от комплексни числа

● x'

```
>> x = [1+3i, 2-2i]
```

```
ans =
```

```
1.0000 + 3.0000i 2.0000 - 2.0000i
```

```
>> x'
```

```
ans =
```

```
1.0000 - 3.0000i
```

```
2.0000 + 2.0000i
```

● $x.'$

```
>> x.'
```

```
ans =
```

```
1.0000 + 3.0000i
```

```
2.0000 - 2.0000i
```



Скаларно произведение

Ако \mathbf{u} и \mathbf{v} са вектори с дължина n , \mathbf{u} е вектор-ред, а \mathbf{v} е вектор-стълб, скаларното произведение на двата вектора е число получено като сума от произведенията на кореспондиращите си елементи

Пример:

```
>> u=[1,2,3];  
>> v=[1;2;3];  
>> prod=u*v  
prod =  
    14
```

на вектори *

$$\underline{\mathbf{u}} = [u_1, u_2, \dots, u_n], \quad \underline{\mathbf{v}} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

$$\underline{\mathbf{u}} \cdot \underline{\mathbf{v}} = \sum_{i=1}^n u_i v_i.$$



Поелементно произведение

Ако **u** и **v** са вектори от един тип, с дължина **n**, поелементното произведение на двата вектора е вектор, елементите на който са произведенията на кореспондиращите си елементи

Пример:

```
>> u=[1,2,3];  
>> v=[1,2,3];  
>> prod=u.*v  
prod =  
    1    4    9
```

на вектори .*

$$\underline{u} \cdot \underline{v} = [u_1 v_1, u_2 v_2, \dots, u_n v_n].$$



Други поелементни операции с вектори

● Деление на вектори ./

- вектори с еднакъв тип и размер

```
>> u./v
```

```
ans =
```

```
1 1 1
```

- деление на нула

```
>> u=[1,0,3];
```

```
>> v./u
```

```
ans =
```

```
1 Inf 1
```

```
>> u./u
```

```
ns =
```

```
1 NaN 1
```



Други поелементни операции с вектори

● Деление на скалар на вектор

```
>> 1./v
```

```
ans =
```

```
1.0000 0.5000 0.3333
```

● Степенуване .^

```
>> v.^2
```

```
ans =
```

```
1 4 9
```



Операции с матрици

•Размер на матрица

```
>> m1=[1,2,3; 4,5,6; 7,8,9;  
0,0,0]
```

```
m1 =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

```
0 0 0
```

```
>> size(m1)
```

```
ans =
```

```
4 3
```

```
>> size(ans)
```

```
ans =
```

```
1 2
```

•Транспониране на матрица

```
>> m2=m1'
```

```
m2 =
```

```
1 4 7 0
```

```
2 5 8 0
```

```
3 6 9 0
```

```
>> size(m1)
```

```
ans =
```

```
4 3
```

```
>> size(m2)
```

```
ans =
```

```
3 4
```



Операции с матрици

• Сума от елементите по стълбове

резултатът е вектор-ред:

```
>> sum(m1)
```

```
ans =
```

```
12 15 18
```

• Сума на елементите от зададен стълб

```
>> sum(m1(1:3,3))
```

```
ans =
```

```
18
```

• Сума на елементите от зададен ред

```
>> sum(m1(3,:))
```

```
ans =
```

```
24
```

• Сума от елементите по редове

резултатът е вектор-стълб

алгоритъм:

- 1) транспониране на матрицата (операция ');
- 2) изчисляване на сумите по стълбове;
- 3) транспониране на резултата.

```
>> sum(m1')'
```

```
ans =
```

```
6
```

```
15
```

```
24
```

```
0
```



Индекси

● Означаване на елемент

- $A(m,n)$ - елементът в ред m и стълб n

- достъп до елемент на матрицата

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

```
ans =
```

```
7
```

- достъп до елемент на масив чрез един индекс

- $A(4,2) = A(8)$

- масивът се съхранява по стълбове като един дълъг вектор-стълб

```
>> A(8)
```

```
ans =
```

```
15
```



Специални матрици

- **zeros** всички елементи са 0;
- **ones** всички елементи са 1;
- **eye** всички елементи са 0, с изключение на тези от главния диагонал, които са 1
- **rand** правилно разпределени случайни числа;
- **randn** нормално разпределени случайни числа;
- **magic** магически квадрат от степен N - квадратна матрица $N \times N$ с елементи естествени числа от 1 до N^2 такива, че сумата от числата във всеки хоризонтал, вертикал или главен диагонал е една съща, равна на $N(N^2+1)/2$;
- **pascal** матрица на Паскал от степен N - симетрична положително дефинирана матрица с целочислени стойности, получени от триъгълника на Паскал (първото и последното число във всеки ред е 1, а другите се получават като сума от двете числа над него)



Примери

```
>> Z=zeros(2,3)
```

```
Z =
```

```
0 0 0
```

```
0 0 0
```

```
>> M=magic(2)
```

```
M =
```

```
1 3
```

```
4 2
```

```
>> P=pascal(4)
```

```
P =
```

```
1 1 1 1
```

```
1 2 3 4
```

```
1 3 6 10
```

```
1 4 10 20
```

на специални матрици

```
>> I=eye(4)
```

```
I =
```

```
1 0 0 0
```

```
0 1 0 0
```

```
0 0 1 0
```

```
0 0 0 1
```



Операции с матрици

- Диагонални матрици
 - само елементите на главния диагонал `diag()` са различни от 0
 - генерират се чрез вектор, който задава елементите на главния диагонал
- Сума от елементите на главния диагонал
- Създаване на огледален образ на матрица
 - вертикален `fliplr()`
 - хоризонтален `flipud()`
- Обратна матрица - `inv()` = 1./

Примери

```
>> d = [-3 -2 -1], D = diag(d)
```

```
d =
```

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

```
D =
```

```
-3  0  0
```

```
 0 -2  0
```

```
 0  0 -1
```

```
>> sum(diag(D))
```

```
ans =
```

```
-6
```

```
>> fliplr(D)
```

```
ans =
```

```
 0  0 -3
```

```
 0 -2  0
```

```
-1  0  0
```

```
>> inv(D)
```

```
ans =
```

```
-0.3333  0  0
```

```
 0 -0.5000  0
```

```
 0  0 -1.0000
```

Преобразуване на матрици

•Завъртане на 90^0

```
>> rot90(m1)
```

```
ans =
```

```
 3  6  9  0
 2  5  8  0
 1  4  7  0
```

•Отделяне на долната лява триъгълна част на матрицата

```
>> tril(m1)
```

```
ans =
```

```
 1  0  0
 4  5  0
 7  8  9
 0  0  0
```

•Премахване на части от матрицата: на съответната част се присвоява символ за празен масив `[]`

•Размерът на матрицата се намалява автоматично

`A(i, :) = []` - премахва i -тия ред на матрицата A ;

`A(:, j) = []` - премахва j -тия стълб на матрицата A ;

`A(:, 3:7) = []` - премахва от 3 до 7 стълб на матрицата A ;

`A(:, [1 5 9]) = []` - премахва 1, 5 и 9 стълб.



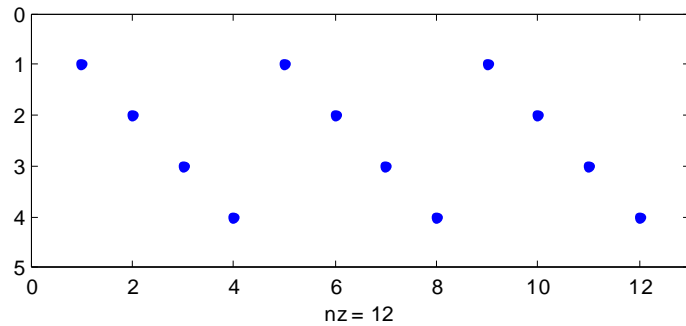
Анализ на елементите

```
>> K=[I,I',diag(1:4)]
```

```
K =
```

```
 1  0  0  0  1  0  0  0  1  0  0  0
 0  1  0  0  0  1  0  0  0  2  0  0
 0  0  1  0  0  0  1  0  0  0  3  0
 0  0  0  1  0  0  0  1  0  0  0  4
```

```
>> spy(K)
```



Операции с матрици

•Извеждане на елемент

```
>> tab(2,3)
```

```
ans =
```

```
1.1683
```

•Изтриване на елементи

- ред
- стълб

•Слепване []

Примери

```
>> tab(2,:)=[]
```

```
tab =
```

```
0 0 0
0.2000 2.2586 2.1521
0.3000 3.1333 2.7961
0.4000 3.7282 2.9987
0.5000 3.9900 2.7279
```

```
>> tab(:,1)=[]
```

```
tab =
```

```
0 0
2.2586 2.1521
3.1333 2.7961
3.7282 2.9987
3.9900 2.7279
```

```
>> tabtab=[tab,tab]
```

```
tabtab =
```

```
0 0 0 0
2.2586 2.1521 2.2586 2.1521
3.1333 2.7961 3.1333 2.7961
3.7282 2.9987 3.7282 2.9987
3.9900 2.7279 3.9900 2.7279
```


Умножение на матрици

- Резултатът от умножение на матрици с размер $m \times n$ и $n \times 1$ е матрица с размер $m \times 1$
- Ако A е матрица с размер $m \times n$ и X е вектор-стълб с размер n , резултатът от умножението е вектор от скаларните произведения на всеки ред от матрицата A с вектора X – вектор-стълб с m елемента.

$$\begin{aligned} Ax &= \begin{bmatrix} 5 & 7 & 9 \\ 1 & -3 & -7 \end{bmatrix} \begin{bmatrix} 8 \\ -4 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 5 \times 8 + 7 \times (-4) + 9 \times 1 \\ 1 \times 8 + (-3) \times (-4) + (-7) \times 1 \end{bmatrix} \\ &= \begin{bmatrix} 21 \\ 13 \end{bmatrix} \end{aligned}$$

```
>> A=[5,7,9; 1,-3,-7];  
>> X=[8;-4;1];  
>> A*X  
ans =  
    21  
    13
```



Умножение на матрица с

$$A = m \text{ rows } \left\{ \begin{array}{c} \boxed{} \\ \boxed{} \\ \vdots \\ \boxed{} \end{array} \right\}, \quad B = \underbrace{\left[\begin{array}{c|c|c} \boxed{} & \boxed{} & \boxed{} \\ \vdots & \vdots & \vdots \\ \boxed{} & \boxed{} & \boxed{} \end{array} \right]}_{p \text{ columns}}$$

$$A[m,n] \times B[n,p] = AB[m,p]$$

$$(m \times \boxed{n} \text{ times } (n \times p) \Rightarrow (m \times p)$$

матрица

```
>> A = [5 7 9; 1 -3 -7]
```

```
A =
```

```
5 7 9
1 -3 -7
```

```
>> B = [0, 1; 2, -3; -4, -5]
```

```
B =
```

```
0 1
2 -3
-4 -5
```

```
>> AB=A*B
```

```
AB =
```

```
-22 -61
22 45
```

```
>> BA=B*A
```

```
BA =
```

```
1 -3 -7
7 23 39
-25 -13 -1
```

Табулиране на функции

Да се табулират функциите

$$y = 4 \sin 3x$$

$$z = 3 \sin 4x$$

за $x = 0; 0:1; 0:2; \dots; 0:5$

```
>> x = 0:0.1:0.5;
```

```
>> y = 4*sin(3*x); z = 3*sin(4*x);
```

```
>> tab = [ x' y' z' ]
```

```
tab =
```

0	0	0
0.1000	1.1821	1.1683
0.2000	2.2586	2.1521
0.3000	3.1333	2.7961
0.4000	3.7282	2.9987
0.5000	3.9900	2.7279

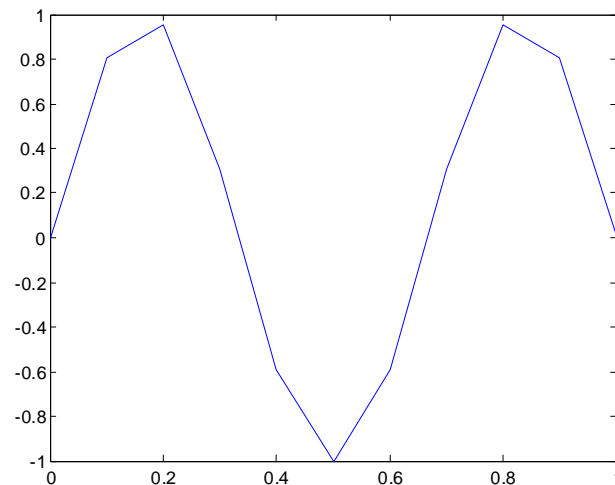


Графика на функция

1. Определяне на функцията
 $y = \sin 3x$
2. Определяне на интервала на изменение на аргументите
 0×1
3. Определяне на стъпката на изменение и броя на итерациите
 $N = 10; h = 1/N; x = 0:h:1;$
4. Определяне на стойностите на аргументите
 $\text{points } x = 0; h; 2h; \dots; 1$

 $\text{linspace}(a,b,n)$,
 a, b – интервал, n – брой
ТОЧКИ

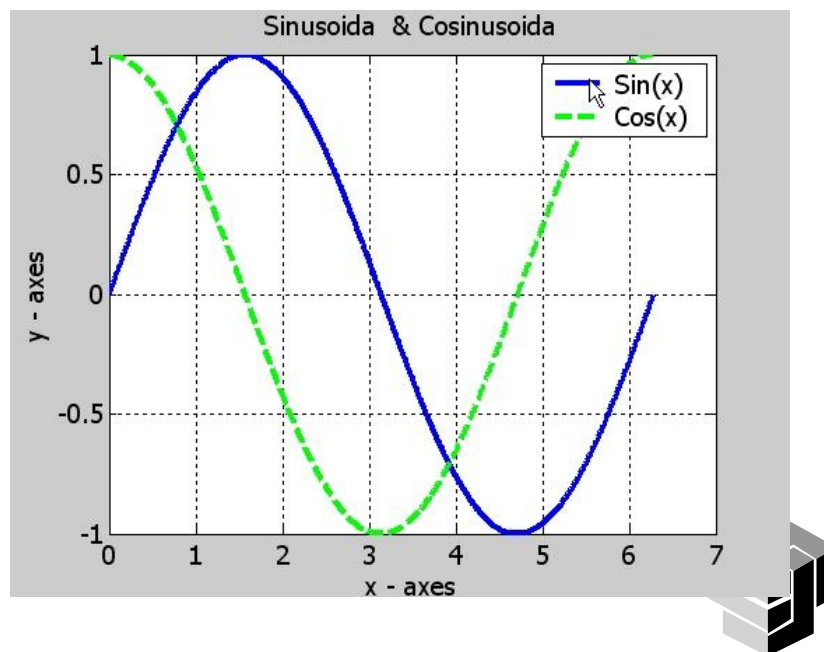
```
>> x = linspace(0,1,11);  
>> y = sin(3*pi*x);  
>> plot(x,y)
```



Графика на функция

1. Определяне на функцията
 $y_1 = \sin x$
 $y_2 = \cos x$
2. Определяне на интервала на изменение на аргументите
 $0 \times 2\pi$
3. Определяне на стъпката на изменение и броя на итерациите
 $h = 2\pi / N = 0.01$
4. Определяне на стойностите на аргументите
points $x = 0; h; 2h; \dots; 2\pi$

```
% интервал и стъпка на аргумента  
x=0:0.01:2*pi;  
% стил, цвят и дебелина на линиите  
plot(x,sin(x),'-b',x,cos(x),'--g','LineWidth',3)  
% шрифт  
set(gca,'FontName','Tahoma','FontSize',14)  
% мрежа  
grid on, xlabel('x - axes'), ylabel('y - axes')  
title('Sinusoida & Cosinusoida')  
legend('Sin(x)','Cos(x)')
```



Графика на функция

