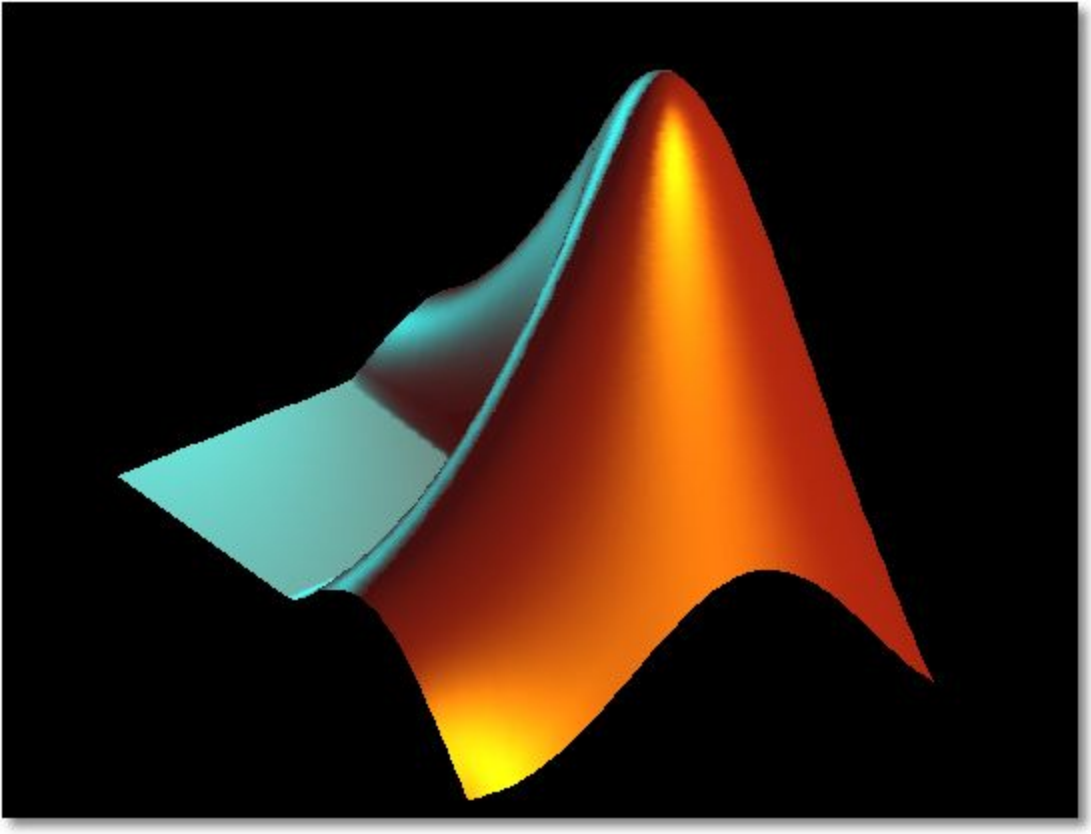


Лекция 2

Вычисления в Matlab

Элементарные функции. Тригонометрические

- sin( )
- cos( )
- tan( )
- cot( )
- asin( )
- acos( )
- atan( )
- acot( )

- sinh( )
- cosh( )
- tanh( )
- coth( )
- asinh( )
- acosh( )
- atanh( )
- acoth( )

- sind( )
- cosd( )
- tand( )
- cotd( )

Имена функций, которые работают со значениями, заданными в градусах, имеют окончание d (от английского слова degree - градус), а у тех функций, которые работают со значениями в радианах, такого окончания нет

Элементарные функции
Экспоненциальные

- exp( )
- log( )
- log10( )
- log2( )
- sqrt( )
- nthroot(x, n)- n-ный корень вещественных чисел

Элементарные функции
Округление и остатки

- **fix** – округление до ближайшего целого в сторону нуля
- **floo**r – округление до ближайшего целого к минус бесконечности
- **ceil** – округление до ближайшего целого к плюс бесконечности
- **round** – округление к ближайшему целому
- **mod(x,y)** – остаток от деления x на y без учёта знака
- **rem(x,y)** – остаток от деления x на y с учётом знака

Примеры.

```
>> b=[1.95  8.17  -4.2];
```

>> fix(b) % округление до ближайшего целого в сторону нуля

```
ans =
 1  8  -4
```

>> floor(b) % округление до ближайшего целого в сторону отрицательной бесконечности

```
ans =
 1  8  -5
```

>> ceil(b) % округление до ближайшего целого в сторону положительной бесконечности

ans =

2 9 -4

>> round(b) % округление до ближайшего целого

ans =

2 8 -4

Элементарные функции Комплексные числа

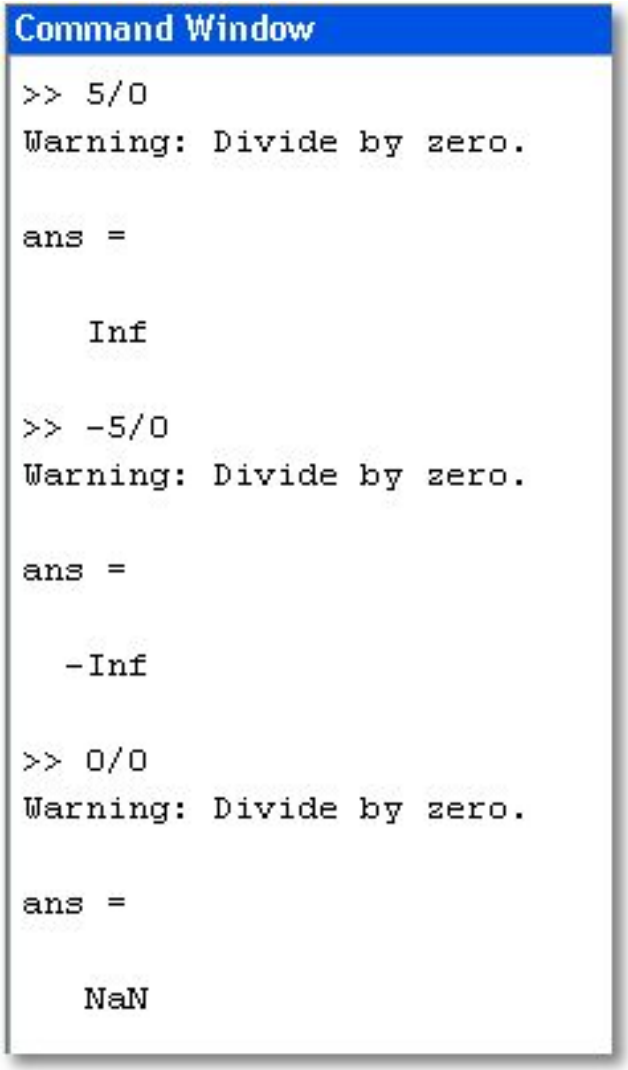
- abs(z) – модуль комплексного числа z
- angle(z) – фаза z (в радианах)
- real(z) – действительная часть z
- imag(z) – мнимая часть z
- conj(z) – комплексно сопряжённое число для z
- complex(a,b) – конструирует комплексное число a+ib
- isreal(z) – возвращает истину, если z – действительное

Элементарные функции

- Просмотреть полный список элементарных функций можно командой
  - `help elfun`

Константы

- pi – число pi
- Inf – бесконечность
- -Inf – минус бесконечность
- NaN (Not a Number) – нечисловое значение



Одномерные массивы

- Задание массива:
  - a = [-3 4 2];
- Диапазоны:
  - b = -3: 2 (b = -3 -2 -1 0 1 2)
  - b = -3:2:5   (b = -3 -1 1 3 5)
- Доступ к элементу:
  - a(3) (будет равно 2)
- Изменение элемента:
  - a(3) = 1
- Количество элементов в массиве: length(a) (будет равно 3)
- Нумерация элементов начинается с 1
- Добавление элементов в массив
  - a(4) = 5;
  - a = [a 5]
- Конкатенация массивов:
  - c = [a b]
- Удаление массива (превращение в пустой массив)
  - a = [ ]

Объединяемые в массив элементы должны отделяться друг от друга либо пробелом, либо запятой.

Примеры.

» V=[l 2 3 4]

V =

1 2 3 4

» sin(V)

ans =

0.8415 0.9093 0.1411 -0.7568

» 3*V

ans =

3 6 9 12

» V^2

??? Error using ==> ^

Matrix must be square

(матрица должна быть квадратной)

» V.^2

ans=

1 4 9 16

» V+2

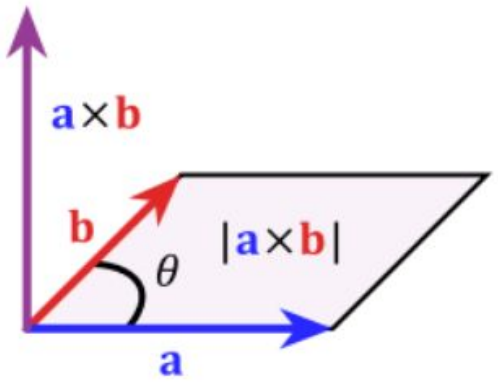
ans =

3 4 5 6

Векторное произведение

Осуществляется функцией **cross(a,b)**, где a и b – векторы, имеющие одну и ту же размерность.

```
» a = [6 5 3]; b= [1 7 6];
» c = cross(a, b)
c =
9 -33 37
```



Скалярное произведение векторов

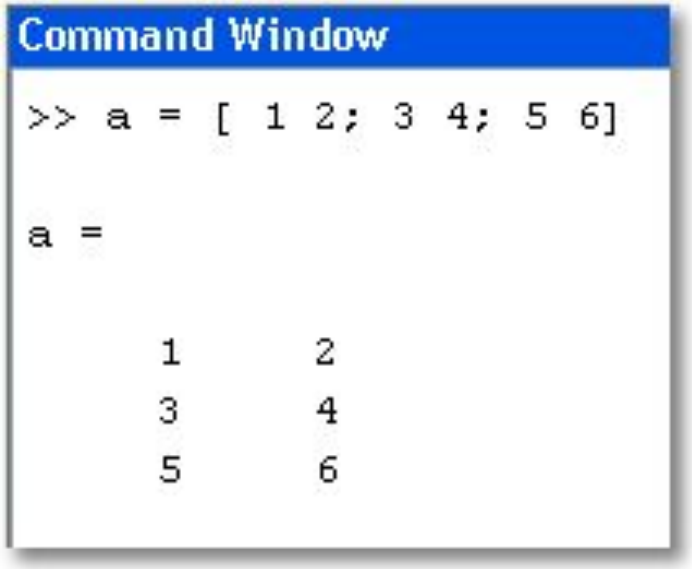
Вычисляется с помощью функции ***sum***. Скалярное произведение равно сумме произведений соответствующих координат

```
>>u=[1 2 3]; v=[3 2 1];
>>sum(u.*v)
ans=
        10
```

Скалярное произведение можно также вычислить как: **u*v’** .

Двумерные массивы

- Задание массива:
  - a = [1 2; 3 4; 5 6];



- Доступ к элементу:

```
>> a(3,1)

ans =

     5

>> a(1,3)
??? Index exceeds matrix dimensions.
```

Обращение к элементам массива производится с помощью круглых скобок:

$a(i)$ – обращение к i-му элементу одномерного массива;

$a(i, j)$ – обращение к элементу двумерного массива, стоящего на пересечении i-ой строки и j-го столбца;

$a(i,:)$ - обращение ко всем элементам i-ой строки двумерного массива;

$a(:,k)$ - обращение ко всем элементам k-ого столбца двумерного массива;

Отсчет индексов в массивах любой размерности начинается с единицы.

Векторы-столбцы и векторы-строки

- Любая строка и столбец матрицы – это вектор
- Векторы, расположенные вдоль строк – векторы-строки (размер 1xn)
- Векторы, расположенные вдоль столбцов – векторы-столбцы (размер nx1)
- К векторам любого типа применима функция length.

  Для массива с функция

  length(c) возвращает число 3. Функция

  не различает вектор-строки и вектор-

  столбцы.

```
>> c = [1; 2; 3]

c =

     1
     2
     3

>> c = [1 2 3]'

c =

     1
     2
     3
```

Размерность и размер матриц

- Размерность массива определяется функцией ndims(A)

```
>> a = [1 2 3; 4 5 6]

a =

     1     2     3
     4     5     6

>> ndims(a)

ans =

     2
```

Массив является двумерным.

- Размер массива – функцией size(A)

```
>> size(a)

ans =

     2     3

>> [m n] = size(a)

m =

     2


n =

     3
```

Конкатенация (склейка)

- Рассмотрим две матрицы

```
>> a = [1 2 3; 4 5 6]

a =

     1     2     3
     4     5     6

>> b = [4 6 7; 0 9 5]

b =

     4     6     7
     0     9     5
```

Конкатенация

- Проведём склейку «в столбик», а затем «в строку»:

```
>> c = [a; b]

c =

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

>> c = [a b]

c =

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

Конкатенация

- При несовпадении размерностей получаем сообщение об ошибке

```
>> c = [c; a]
??? Error using ==> vertcat
CAT arguments dimensions are not consistent.
```

Пример. Создать матрицу $M = \begin{pmatrix} A, B \\ C, D \end{pmatrix}$

```
>> A=[1,2,3; 4,5,6]; % задание матрицы A
>> B=[11;12];        % задание матрицы B
>> C=[7,8;9,10];     % задание матрицы C
>> D=[13,14;15,16];  % задание матрицы D
>> M=[A,B;C,D]  % формирование матрицы M
% результат
```

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

Диапазоны

- Можно использовать как для задания значений векторов, так и для задания диапазонов индексации
- Рассмотрим другие примеры

```
>> a = magic(5)

a =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> a(2:3,4:5)

ans =

    14    16
    20    22
```

Диапазоны

В качестве индексов могут выступать векторы, содержащие номера нужных строк и столбцов.

```
b =
     4     3    -1
     2     7     0
    -5     1     3
>> i=[1 2];
>> j=[2 3];
>> b(i,j)
ans =
     3    -1
     7     0
```

Диапазоны

```
>> a = magic(5)

a =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> a(3,:)

ans =

     4     6    13    20    22

>> a(:,1)

ans =

    17
    23
     4
    10
    11
```

```
>> a = magic(3)

a =

     8     1     6
     3     5     7
     4     9     2

>> a(:)

ans =

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

Диапазоны

- Для обращения к последнему элементу любой размерности можно использовать служебное слово `end`:

```
>>A(2,end-1)
% Возвращение предпоследнего
   элемента втрой строки матрицы
ans=
      14
```

```
>> A = magic(5)

A =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> B = A(:,3:end)

B =

     1     8    15
     7    14    16
    13    20    22
    19    21     3
    25     2     9
```

Удаление строк и столбцов

```
>> a = magic(5)

a =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> a(2:3,:) = [ ]

a =

    17    24     1     8    15
    10    12    19    21     3
    11    18    25     2     9
```

```
>> a = magic(5)

a =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> a(:,3:4) = [ ]

a =

    17    24    15
    23     5    16
     4     6    22
    10    12     3
    11    18     9
```

Перестановка элементов

```
>> b = 1:3:11

b =

     1     4     7    10

>> b([4 2 1 3])

ans =

    10     4     1     7
```

```
>> a = magic(5)

a =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

>> a = a(:,[3 5 2 4 1])

a =

     1    15    24     8    17
     7    16     5    14    23
    13    22     6    20     4
    19     3    12    21    10
    25     9    18     2    11
```