

Работа в командном окне

(или сразу работаем в текстовом редакторе)

Системные переменные

eps – бесконечно малое, обусловленное точностью (2.2204e-016)

realmax – наибольшее положительное вещественное число (1.7977e+308)

realmin – наименьшее положительное вещественное число (2.2251e-308)

intmax - наибольшее положительное целое число (2147483647)

intmin - наименьшее положительное целое число (-2147483648)

ans – системная переменная (результат последней операции, в отсутствии результирующей переменной)

pi - 3.141592653589793 (format long)

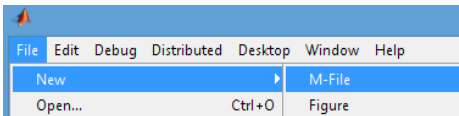
inf – бесконечно большое

NaN (not a number) – нечисловой формат

exp(1) (2.718281828459046) нет зарезервированной системной переменной

end – последний элемент

Простейшие операции



Задание 1. Выполните вычисления:

1+2

3-4

3*5

10/3

3^3

format rational

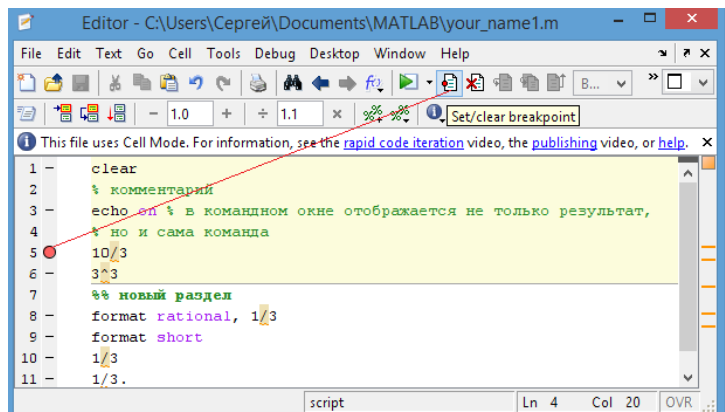
1/3

format short

1/3, 1/3.

1/0 % результат - системная переменная

0/0 % нечисловой формат - системная переменная



1

Задание 2

Задайте векторы:

d1=[1 1 1], d2=[2 2 2]

d3=[5; 5; 5], d4=d2'

Выполните операции

d1+d2, d4+d3,

d1*d2, d4*d2,

d1/d2, d2/d1

(объясните результат)

Замечание. Воспользуйтесь стеком и редактированием текущей строки, если это целесообразно.

Задание 3

Выполняйте и анализируйте ∇ результат

a='MatLab forever'

(результаты заданий 1-2 остаются в сессии)

who

whos % почему в **a** такое количество байт?

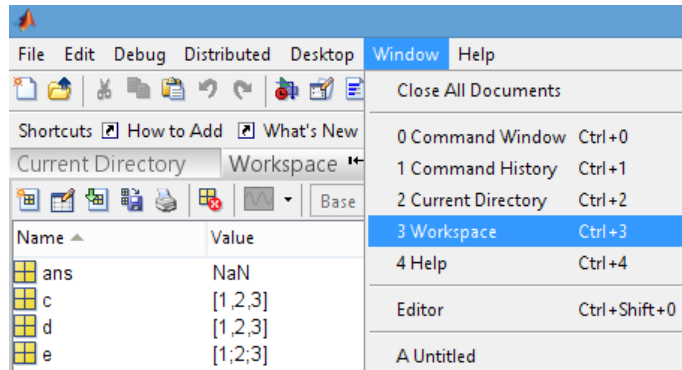
clear d2

whos

clear d1,d3 % объясните результат

who

clear



Панель Workspace – содержит информация о глобальных переменных (команда **who** – интерактивный аналог); **whos** - расширенная информация о данных (+ какие) типы данных (классы)

Конструкторы матриц и операции с векторами и матрицами

Задание 4. Матрицы из нулей (конструктор **zeros**)

Выполните команды:

zeros(5) % создается в памяти матрица пятого порядка из нулей

zeros(2,3)

zeros(4,3,5)

A=[1,2,3;4,5,6]

B= zeros(size(A))

2

Задание 5. Матрицы из единиц и единичная

Выполните команды:

ones(5)

ones(2,3)

ones(4,3,5)

A=[1,2,3;4,5,6]

eye(5) % единичная матрица

B=eye(size(A))

Задание 6. Случайные матрицы

rand(5) % матрица 5-го порядка, заполненная случайными числами, равномерно распределенными на интервале (0, 1)

randn(10,2) % матрица заданного порядка, случайные числа генерируются с помощью нормального распределения нулевым средним и единичной дисперсией

randint(3,3,[3 9]) % генерируется случайно матрица третьего порядка, элементы которой целые числа не меньше 3-х и не больше 9-ти. В некоторых версиях ML используется короткая команда **randi** (читайте справочную систему **help randi**)

Задание 7. Выбор элементов матрицы, редактирование

Задайте матрицы:

A=[1 2 3; 4 5 6; 7 8 9]; % заметим, что ; - подавляет вывод результата

B=[1 1 1; 2 2 2; 3 3 3];

Для доступа к элементам массива используются круглые скобки $A(2,3)$ – элемент 2-й строки и 3-го столбца; $B(1:3)$ - выбран столбец [1;2;3], $B(2,:)$ – вторая строка
Выберите 1-й столбец матрицы A и 2-й столбец матрицы B.

Выберите вторую и третью строку матриц.

Замечание. В памяти ML любая матрица хранится как вектор-строка, построенная последовательно из её векторов-столбцов. Наличие одинарной нумерации предполагает такое соответствие.

A=rand (6)

AL=tril(A), tril(A,-2)

AU= triu(A), triu(A,2)

AD=diag(A), v=diag(A,3),

B=diag(v), diag(A),

Факультативно:

fliplr(A) – поворот матриц вправо-влево

flipud(A) – поворот матриц вверх-вниз

rot90(A) – поворот на 90 градусов

Задание 8. Операции с матрицами

Выполните операции с матрицами

A+B, A-B, A*B

% операции с точкой – векторные операции, выполняются поэлементно

A.*B % поэлементное умножение

A.*10 % умножить на число; заметим, что для умножения на скаляр не обязательно пользоваться векторной операцией, достаточно **A*10**

A.^2 %(каждый элемент матрицы, возведен в степень 2)

A^2 %(матрица, умноженная сама на себя (только для квадратных матриц!))

A' % транспонирование ли матрицы?

inv(A) – обратная матрица к A или равносильно:

A^(-1)

sum(A), sum(A')

sum(diag(A)), sum(diag(fliplr(A)))

sum(sum(A)), prod(A)

max(max(A))

Задание 9. Справочная система

Выполните команды:

help elfun % Познакомьтесь со справкой по элементарным математическим функциям

help matfun % по изучению свойств матриц и специальных операций
help specfun % запомните, как вызывать справку по специальным функциям.
Выполните команду `help sum`. Объясните, как работает функция `sum`.

Операции сравнения и логические операции

Задание 10

Скопируйте выражения в командную строку и выполните проверку:

`123+456==579`

`3>9`

`[3 7 4]>[3 2 3]` % поэлементное сравнение

`t=ans`

`all(t)` % если в `t` все логические единицы, то результат `true` (или 1)

`any(t)` % если в `t` есть хотя бы одна логические единицы, то результат `true` (или 1)

`2<2.5000`

`7*8>=56`

`2^32<=65536`

`X=[1 2 3]`

`Y=['1','2','3']`

`Z=['1','2','A']`

Выполните следующие операции сравнения

`X==Y`

`X<Y`

`Z==Y`

Как происходит сравнение?

Задание 11. Логические операции с массивами

Задайте массивы:

`P=[1 1 1]`

`Q=[0 1 0]`

`R=[0 0 1]`

Выполните следующие логические операции

`Q|R`

`P&R`

`~Q`

Посмотрите расширенную справку по логическим операциям, операциям отношения, операциям над множествами, для этого в командной строке наберите

help > (знак больше)

Придумайте примеры операций над множествами.

Замечание. Язык нестрогой типизации обуславливает необходимость постоянного, насколько это требует логика программы, контроля типов.

В статье контекстного поиска следует набрать `is*` в закладке, указано ниже

Enter index item: is*

Получите список всех функций контроля типов, например,
`isequal` – истина, если аргументы - матрицы равны поэлементно
`isscalar` – истина, если аргумент скаляр.
`isvector` – истина, если аргумент вектор.
`Isrow` – истина, если аргумент строка.
`Iscolumn` - истина, если аргумент столбец.
`ismatrix` – истина, если аргумент матрица.

Задание 12. Объединение матриц

Задайте матрицы `r1` и `r2` одинакового размера.

Объедините их двумя способами:

`[r1;r2]` `[r1 r2]`

Изучите справку для команды `cat` и с её помощью выполните оба предыдущих задания.

Задание 13. Элементарные математические функции

Задайте комплексные числа и найдите их характеристики:

`d1=5+4i`

`d2=2-3j`

`abs(d1)` % модуль числа

`real(d1)` % вещественную часть

`imag(d1)` % мнимую часть

`conj(d2)` % комплексно-сопряженное, сравните с

`d1'`

Задайте вектор вещественных `x` и по результатам сформулируйте правила работы функций округления:

`ceil(x)`, `fix(x)`, `floor(x)`, `round(x)`

проверьте работу функций

`mod(x,y)`, `rem(n,m)`

сравните результаты. Уточните назначение этих функций с помощью `Help`.

Задание 14. Задание последовательностей (векторов-строк)

Выполните команды:

`q=2:2:8` % начальное значение: шаг: конечное значение

`r=3:3:12`

`v=q+r`

`f=v^2`

`q=4:end`

Элементы программирования. Скрипты, процедуры и функции.

*Скрипт файл (script) – внешний *.m файл, состоящий из последовательности ML-команд. Все переменные глобальные, в качестве первой команды следует использовать clear*

Задание 15. Скрипт файл

Создайте script. Задайте две квадратные матрицы. Для этих матриц выполните основные операции:

A+B, A-B, A*B, A/B, A\B, A.*B, A.^B

Запустите программу.

Оцените результат.

Задание 16. Скрипт файл

Создайте script построения графика функции.

t=-pi:0.3:pi % изменение аргумента функции

e=sin(t) % массив значений функции

plot(t,e) % построение графика функции

Задание 17. Процедура-функции

Создайте процедуры-функции, которые вычисляют:

1. Максимальное значение элементов главной диагонали матрицы.
2. Сумму всех элементов матрицы, вне главной диагонали.

Замечание. Первая строка функции начинается с заголовка. Заголовок функции должен иметь один выходной параметр. Имя файла и имя функции совпадают и начинаются с буквы. Синтаксис заголовка-функции следующий:

```
function x=namefunction(parameter1, par2)
```

% комментарий о назначении входных и выходных параметров

Задание 18. Процедура

Создайте две процедуры, которые вычисляют:

1. $n!$, $1/n!$ (используем функцию factorial)

Файл **myf.m**

вызов: [p, r]= myf(10)

2. Корни квадратного уравнения

Файл **sreq.m**

вызов: [x1, x2]= sreq(1,9,6)

Замечание. Заголовок процедуры аналогичен функции, но с любым количеством выходных параметров

```
function [x,y]=namefunction(parameter1, parameter2)
```

% комментарий о назначении входных и выходных параметров