**Курс лекций и практических занятий MATLAB (Рабочий вариант)**

**Занятие №1**

**Общие принципы работы MATLAB**

1. Математический пакет MATLAB. Общая характеристика пакета.
2. Интерфейс (среда) MATLAB (окна, меню).
3. Работа в командной строке. Формат.
4. Операторы и операции.
5. Сохранение рабочей среды.
6. Встроенные элементарные функции.

**MATLAB** – интерпретатор, то есть операторы выполняются без предварительной компиляции. **MATLAB** имеет собственную оболочку и язык для разработки и выполнения программ.

По умолчанию открыто три (основные) окна:

* Command Window – предназначено для ввода команд.
* Workspace – отображает переменные, хранящиеся в памяти.
* Current Folder – текущая дирректория.

Операторы MATLAB вводятся в командном окне или из файлов с именами \*.m

MATLAB – расширяемая система. Оператор-> .m файл-> пакет расширения.

**Переменные и знаки операций**

Переменные в **MATLAB** не объявляются, а вводятся по ходу выполнения программы. Имена назначаются пользователем или системой и формируются как произвольные последовательности латинских букв и цифр (допустим также символ подчеркивания), начинающиеся обязательно с буквы. *При этом система различает прописные и строчные буквы*.

По умолчанию все переменные являются матрицами из вещественных чисел типа **double**.

**Замечание:** Если после введенной команды не поставить точку с запятой (**;)**, то результат выводится на консоль.

**Замечание:** Если необходимо продолжить ввод данных на другой строке, то ставится без пробелов три точки +Enter.

**Замечание:** Повторение команды – стрелка вверх ↑ (вниз ↓).

По умолчанию результат помещается в переменную с именем **ans**.

Размерность массивов в **MATLAB** произвольная, т.е. можно задавать массивы 3-x мерные,

4-x мерные и т.д.

В системе по умолчанию определены некоторые переменные, основными из которых являются следующие:

**i** или **j** – мнимая единица;

**pi** – число 3.141592653589793;

**eps** – машинная точность (2^-52);

**realmin** – наименьшее число с плавающей точкой на данном компьютере;

**realmax** – наибольшее число с плавающей точкой;

**inf** – значение машинной бесконечности;

**NaN** – указание на нечисловой характер данных (Non-a Number);

**Операции**:

**+** сложение,

**-** вычитание,

**\*** матричное умножение,

**/** матричное деление,

**^** возведение в степень.

**Формат данных:**

**format** long (short)

>> z=i+j

z =

0.000000000000000 + 2.000000000000000i

>> z1=(i+j)'

z1 =

0.000000000000000 - 2.000000000000000i

**Замечание**: Для комплексного сопряжения используется апостроф после скобок с выражением.

**Некоторые команды для управления окном командного режима**

* **home** – поместить курсор в левый верхний угол командного окна;
* **clc** – очистить командное окно и поместить курсор в левый угол командного окна;
* **who** – получить список переменных, хранящихся в памяти;
* **whos** – список переменных с дополнительной информацией;
* **clear all** – удалить все переменные из памяти;
* **clear var** – удалить переменную var;
* **clear global** – уничтожение всех глобальных переменных;
* **help** – получить помощь;
* **help имя\_команды** – получить помощь по конкретной команде;
* **format** – управляет форматом вывода переменных на консоль;
* **pause** – прерывание работы программы
* **%** – текстовый комментарий,

**.mat** – бинарные файлы;

**.m** – текстовый файл (команды, функции);

**.c** – программа по Си;

**.mex** – откомпилированный код.

Сеанс работы в MATLAB принято называть сессией.

**Сохранение рабочей среды. MAT файлы**

Самый простой способ сохранить все значения переменных — использовать в меню **Variable** пункт **Save Workspace**. При этом появляется диалоговое окно **Save Workspace**, в котором следует указать каталог и имя файла. По умолчанию предлагается сохранить файл в подкаталоге **work** основного каталога **MATLAB.** Программа сохранит результаты работы в файле с расширением **mat**. Теперь можно закрыть **MATLAB**. В следующем сеансе работы для восстановления значений переменных следует открыть этот сохраненный файл при помощи подпункта **Open** меню **File**. Все переменные, определенные в прошлом сеансе, опять стали доступными. Их можно использовать во вновь вводимых командах.

**Сохранение рабочей сессии:**

**save** fname файл fname.mat

Загрузка рабочей области ранее проведенной сессии:

**load** fname (загрузка .mat файла).

Выход из системы: **quit, exit, Ctrl+Q**.

В MATLAB имеется возможность записывать исполняемые команды и результаты в текстовый файл (вести журнал работы), который потом можно прочитать или распечатать из текстового редактора. Для начала ведения журнала служит команда **diary**. В качестве аргумента команды **diary** следует задать имя файла, в котором будет храниться журнал работы. Набираемые далее команды и результаты их исполнения будут записываться в этот файл.

**Пример:**

**diary example1.txt**

**x=5**

**x =**

**5**

**y=10**

**y =**

**10**

**z=x\*y**

**z =**

**50**

**save work1**

**diary off**

**quit**

Просмотреть файл **example1.txt.** При дальнейшем использовании данные добавляются в файл!

**diary on**

**Работа с HELP**

**Вкладка Resources->Help->Documentation->MATLAB**

**Встроенные элементарные функции**

1. Тригонометрические.
2. Гиперболические.
3. Экспоненциальные, логарифмические.
4. Для работы с комплексными числами.
5. Округление.

**help elfun – вызов справки по элементарным функциям.**

**Тригонометрические функции:**

**sin, cos, tan, cot – синус, косинус, тангенс, котангенс.**

**sec, csc – секанс, косеканс.**

**asin, acos, atan, acot – арксинус, арккосинус, арктангенс, арккотангенс.**

**acec, acsc – арксеканс, арккосеканс.**

**Замечание:** Аргументы функций записываются в радианах.

**Гиперболические функции:**

**sinh, cosh, tanh, coth – гиперболические синус, косинус, тангенс, котангенс.**

**sech, csch – гиперболические секанс, косеканс.**

**asinh, acosh, atanh, acoth – гиперболические арксинус, арккосинус, арктангенс, арккотангенс.**

**acech, acsch – гиперболические арксеканс, арккосеканс.**

**Экспоненциальные, логарифмические и степенные функции:**

**exp – экспоненциальная;**

**log – натуральный логарифм;**

**log10 – десятичный логарифм;**

**log2 – логарифм по основанию 2;**

**pow2 – возведение числа 2 в степень;**

**sqrt – квадратный корень.**

**Функции для работы с комплексными числами**

**real – вещественная часть комплексного числа;**

**imag – мнимая часть комплексного числа;**

**abs – модуль комплексного числа;**

**angle – угол (в радианах) комплексного числа;**

**complex – формирует комплексное число по вещественной и мнимой частям.**

**Пример:** % Проверить работу функций, используя комплексное число

**z=1+2i**

**real(z)**

**imag(z)**

**abs(z)**

**angle(z)**

**complex(3, 4)**

**Функции округления и остаток от деления**

**fix - округление до ближайшего к 0 целого (усечение дробной части); fix(1.7), fix(-1.7) => (1,-1).**

**floor - округление до ближайшего целого по направлению -∞ (до меньшего целого);floor(2.7), floor(-2.4)=> (2,-3).**

**ceil - округление до ближайшего целого по направлению +∞ (до большего целого); ceil(2.4), ceil(-2.8)=>(3,-2).**

**round - округление до ближайшего целого; round(-3.5) =>-4.**

**mod - остаток по модулю деления (знак второго аргумента); mod(-5,2) => 1.**

**rem - остаток от деления (знак первого аргумента); rem(-5,2) => - 1.**

**sign - знак числа; sign(-3) => -1.**

**Замечание:** Функция **mod()** в MATLAB работает **не так, как ожидает пользователь.**

**Пример:** Согласно реализации функции в **MATLAB: mod(x,y) ==x-y\* floor(x/y)**

Пусть **x= -7, y=3**. Тогда, **mod(-7,3)=-7-3\*floor(-7/3)=-7-3\*(-3)=-7+9=2.**

Правильный ответ должен быть:  **1.**

**Функция rem() использует функцию fix() и работает правильно (как заявлено)!**

**Массивы**

**Векторы и матрицы**

Для формирования векторов и матриц используются символы **[]**.

Элементы массива вводятся через пробел или запятую.

Строки отделяются друг от друга символом **;** (точка с запятой).

Таким образом, [,] [;] – объединение массивов.

**Примеры:**

**a=[1 2 3]** – вектор-строка;

**b=[1;2;3]** – вектор-столбец.

**C= [4,5,6; 7,8,9]** – матрица, размерностью 2х3.

**Конкатенация матриц** (размерности должны быть согласованы)

**[A,B,C]** – горизонтальное объединение массивов.

**[A;B;C]** – вертикальное объединение массивов.

**Пример:**

**D=[C; a]** – матрица, размерностью 3х3.

**E=[D,b]** – матрица, размерностью 3х4.

Обращение к элементу матрицы **D**, расположенному на пересечении строки с номером **i** и столбца с номером **j**, имеет вид **D(i, j)**.

**Блоки матриц**

Блок матрицы можно получить, используя символ перечисление:

Вектор-столбец с номером **j** – **D(:,j)**;

Вектор-строка с номером **i** – **D(i,:)**.

**Пример:**

**D(:,2)** % второй столбец матрицы D;

**D(2,:)** % вторая строка матрицы D.

Оператор **start:step:fin** задает арифметическую прогрессию.

Первый член равен значению **start**, шаг прогрессии – **step**, а последний – ближайший к **fin** член прогрессии, принадлежащий сегменту **[start; fin].**

В записи **start:fin**, шаг по умолчанию равен 1.

**Пример:**

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

**B=A(1:3,2:3)**

**B =**

**2 3**

**5 6**

**8 9**

**B=A(1:1.7:3,2:3)**

**B =**

**2 3**

**8 9**

**Формирование матриц из блоков других матриц**

Пусть дана матрица **A(4,4)**.

**Задание 1:** Построить матрицу **B**, составленную из блоков матрицы A. Пусть она будет состоять из двух блоков. Первый блок – элементы матрицы A, стоящие на пересечении первых двух строк и первых двух столбцов. Второй блок – элементы матрицы A, стоящие на пересечении второй и третьей строк и третьего и четвертого столбца.

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

**Индексация**

Подматрицу исходной матрицы можно получить, указав вместо индексов векторы, содержащие номера строк и столбцов исходной матрицы.

**Пример:**

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

**u=[1 2];**

**v=[1 2];**

**u1=[2 3];**

**v1=[3 4];**

**B=[A(u,v), A(u1,v1)]**

**B =**

**1 2 7 8**

**5 6 11 12**

**Удаление строк и столбцов**

Для удаления строки (столбца) надо присвоить строке (столбцу) пустой массив [].

**Пример:**

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

**A(2,:)=[]**

**A =**

**1 2 3**

**7 8 9**

**Функции работы с массивами**

**length(X)** – длина вектора.

**size(X)** – размер вектора.

**dot(a,b) –** скалярное произведение

**cross(a,b)** – векторное произведение.

**zeros(m,n)** – формирование массива нулей (m,n).

**оnes(m,n)** – формирование массива единиц.

**eye(n)** – формирование единичной матрицы.

**rand(n)** – формирование матрицы (n,n) случайных чисел.

**diag(a)** – формирование диагональной матрицы, по диагонали элементы вектора **a.**

**blkdiag()** – формирование блочно-диагональной матрицы.

## Создание матриц специального вида

Заполнение прямоугольной матрицы нулями производится встроенной функцией **zeros**

**Z=zeros(4,5)**

**Z =**

**0 0 0 0 0**

**0 0 0 0 0**

**0 0 0 0 0**

**0 0 0 0 0**

Единичная матрица создается при помощи функции **eye**

**I=eye(4,5)**

**I =**

**1 0 0 0 0**

**0 1 0 0 0**

**0 0 1 0 0**

**0 0 0 1 0**

Матрица, состоящая из единиц, образуется в результате вызова функции **ones**

**E=ones(3,4)**

**E =**

**1 1 1 1**

**1 1 1 1**

**1 1 1 1**

**R=rand(5)**

**R =**

**0.3998 0.1818 0.5797 0.3510 0.1233**

**0.2599 0.2638 0.5499 0.5132 0.1839**

**0.8001 0.1455 0.1450 0.4018 0.2400**

**0.4314 0.1361 0.8530 0.0760 0.4173**

**0.9106 0.8693 0.6221 0.2399 0.0497**

## Основные матричные операции

При использовании матричных операций следует помнить, что для сложения или вычитания матрицы должны быть одного размера, а при перемножении число столбцов первой матрицы обязано равняться числу строк второй матрицы. Сложение и вычитание матриц, также как чисел и векторов, осуществляется при помощи знаков плюс и минус.

**+ сложение,**

**- вычитание,**

**\* матричное умножение,**

**\ матричное деление** % X=A\B эквивалентно X=inv(A)\*B 

**/ обратное (справа налево) деление матриц** % эквивалентно X=A\*inv(B) 

**^ возведение в степень.**

**Пример: % сложение и умножение матриц, умножение на скаляр**

**A= [1 5; 3 6]**

**B= [ 2 4; 1 6]**

**C=A+B**

**D=A-B**

**K=5+A**

**E=5\*A**

**F=A\*5**

**G=A\*B**

**H=B\*A**

**I=A^2**

**Операторы (поэлементное действие)**

**' транспонирование;**

**.\* поэлементное умножение;**

**./ поэлементное деление;**

**.\ обратное поэлементное деление;**

**.^ поэлементное возведение в степень.**

**Проверить работу «поэлементных операций»**

**A= [1 5; 3 6]**

**B= [ 2 4; 1 6]**

**C=A.\*B**

**D=A./B**

**E=A.\B**

**F=A.^B**

**Пример:** % для матриц деление слева – \, и деление справа /

**A= [1 5; 3 6]**

**B= [ 2 4; 1 6]**

X=A\B % эквивалентно X=inv(A)\*B 

X=A/B % эквивалентно X=A\*inv(B) 

**Задание:** Решить СЛАУ и проверить результат

A=[2 3 1;3 1 -1;2 -1 3] , b=[11;2;9]

**Задание: найти скалярное произведение «вручную» и, используя MATLAB.**

**a=[1 2 3]**

**b=[3 2 1]**

**c=sum(a.\*b)**

**c1=a\*b'**

**c2=dot(a,b)**

**Векторное произведение**

**Задание: найти векторное произведение «вручную» и, используя MATLAB.**

**a=[1 2 3]**

**b=[3 2 1]**

**v=cross(a,b)**