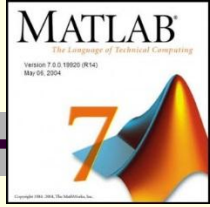


Курс: Программные продукты в математическом моделировании.

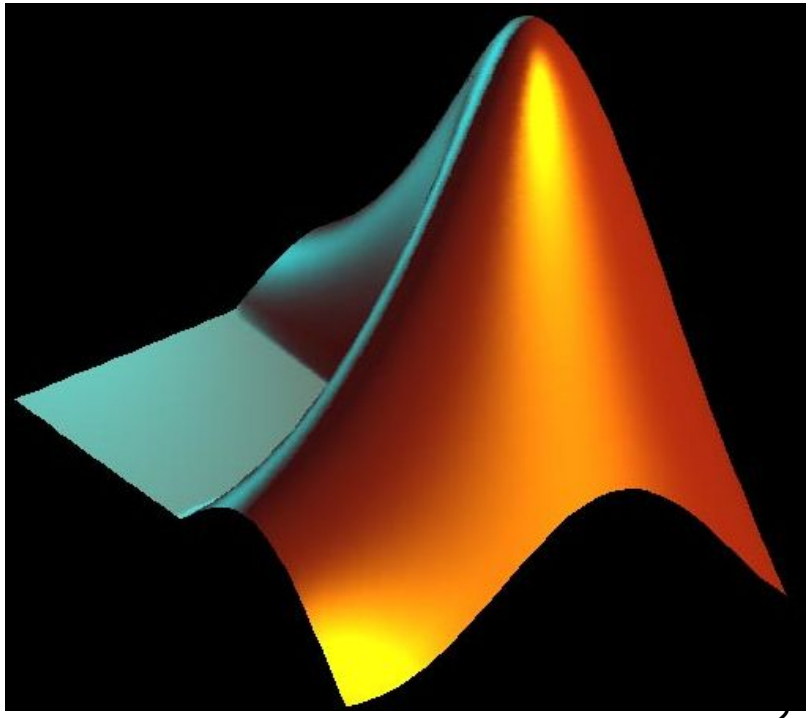
Лекция

Операции с матрицами в MatLab 7

Возможности системы МатЛаб

MATLAB- это уникальная коллекция реализаций современных численных методов компьютерной математики.

- Вычисления,
  визуализация,
  программирование

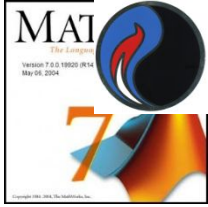


MATLAB - уникальная коллекция реализаций современных вычислительных методов

- матричные и логические операторы
- элементарные и специальные функции
- полиномиальная арифметика
- многомерные массивы, массивы записей и ячеек
- дифференциальные уравнения
- вычисление квадратур
- поиск корней нелинейных алгебраических уравнений
- оптимизация функций нескольких переменных
- одномерная и многомерная интерполяция
- аналитические расчёты

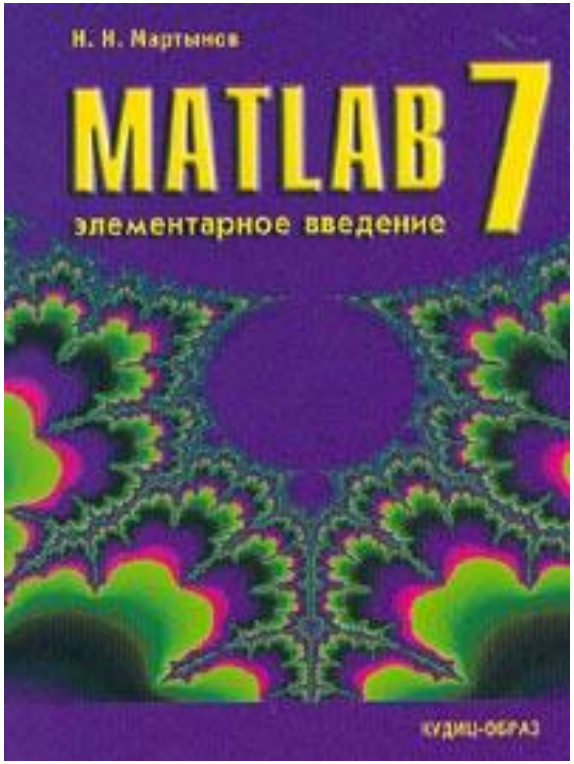
и многое другое

Возможности системы MATLAB



- **В области визуализации и графики**:
  - возможность создания двумерных и трехмерных графиков
  - осуществление визуального анализа данных
- **В области программирования**:
  - интерактивная среда программирования
  - язык программирования, близкий к обычной математической нотации
  - свыше 1000 встроенных математических функций
  - работа с текстовыми и двоичными файлами
  - применение программ, написанных на Си, C++, ФОРТРАН и JAVA
- **Средство построения графического интерфейса пользователя** (GUI)
  - облегчает взаимодействие пользователя с системой

Список рекомендуемой литературы :



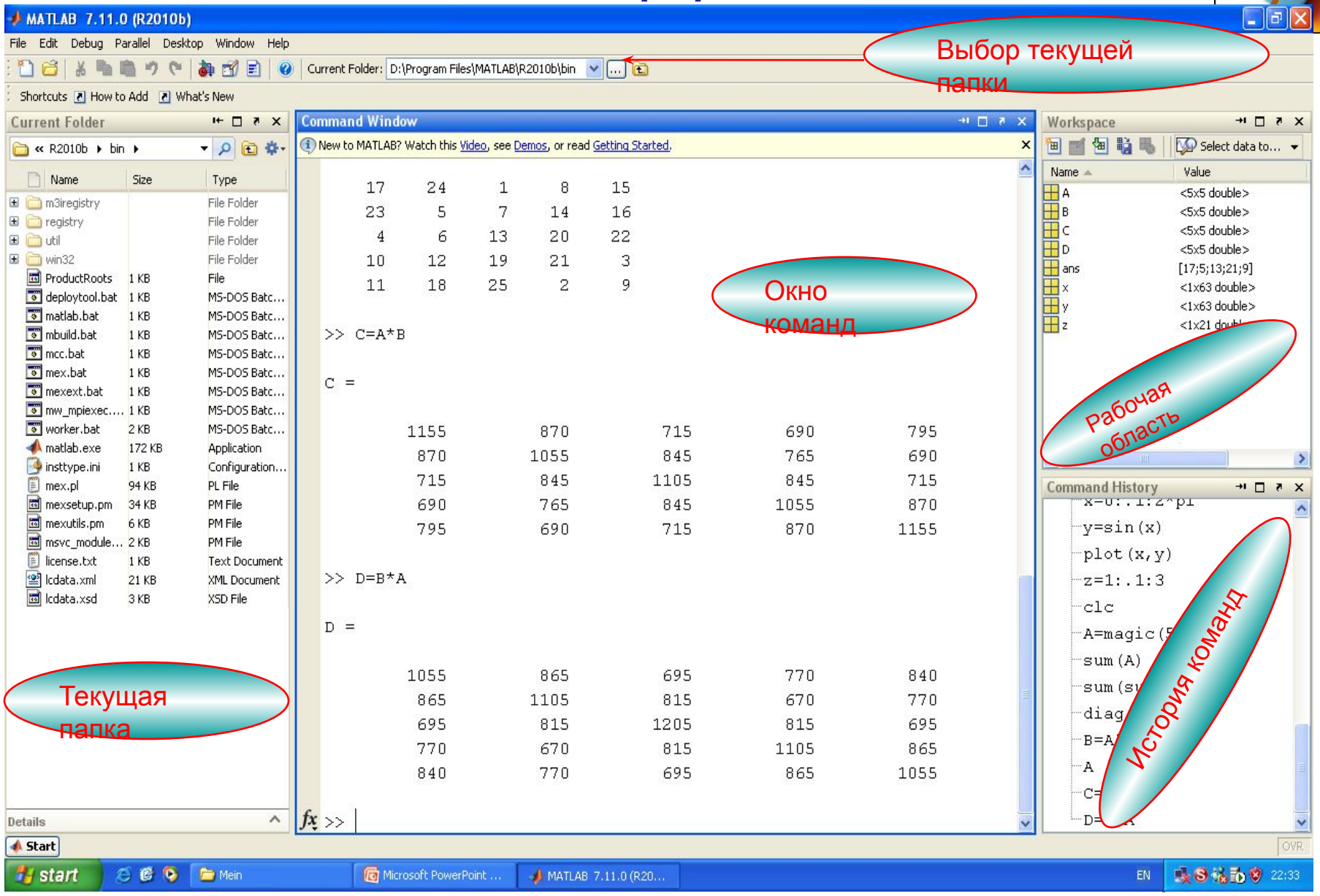
Мартынов Н.Н. Matlab 7. Элементарное введение. -М:КУДИЦ-ОБРАЗ,2005.-416с

Список рекомендуемой литературы :

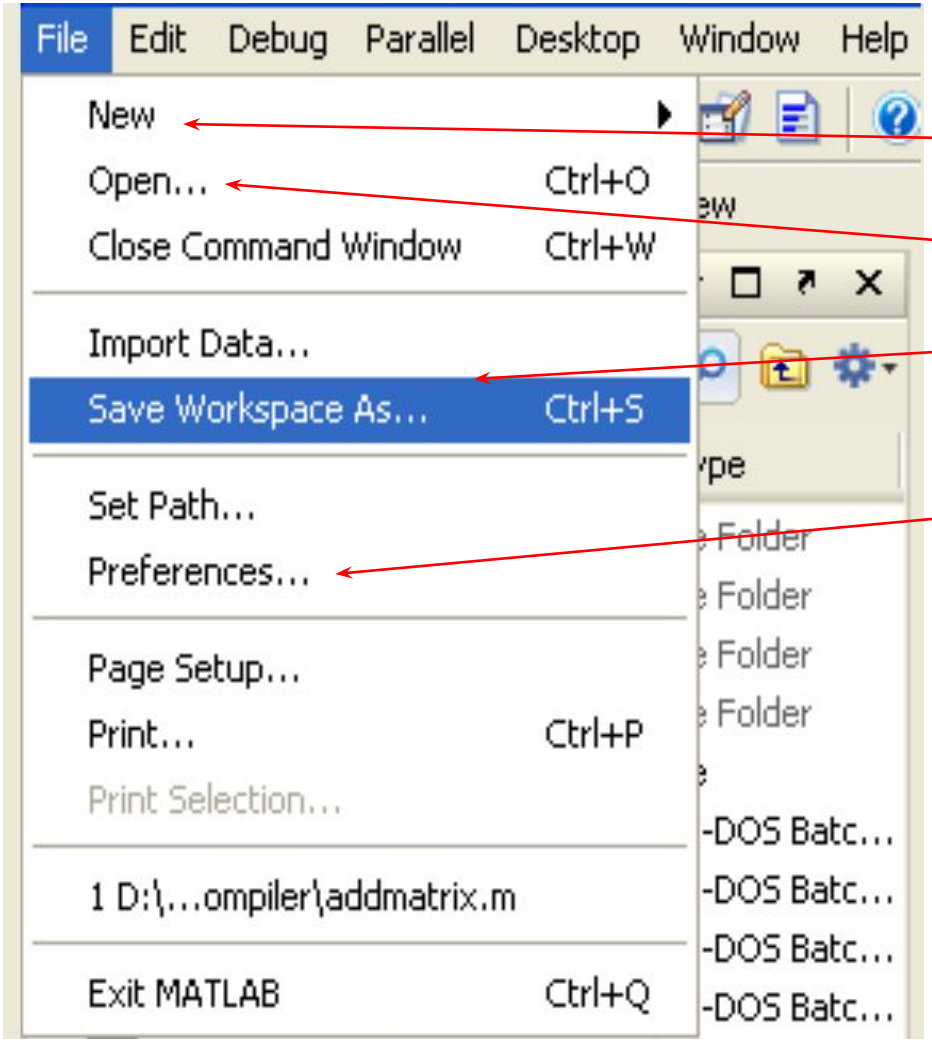
Юлий Кетков, Александр Кетков, Михаил Шульц

Matlab 7. Программирование, численные методы. -БХВ-Петербург, СПб ,2005.-742 с

Основной интерфейс MATLAB

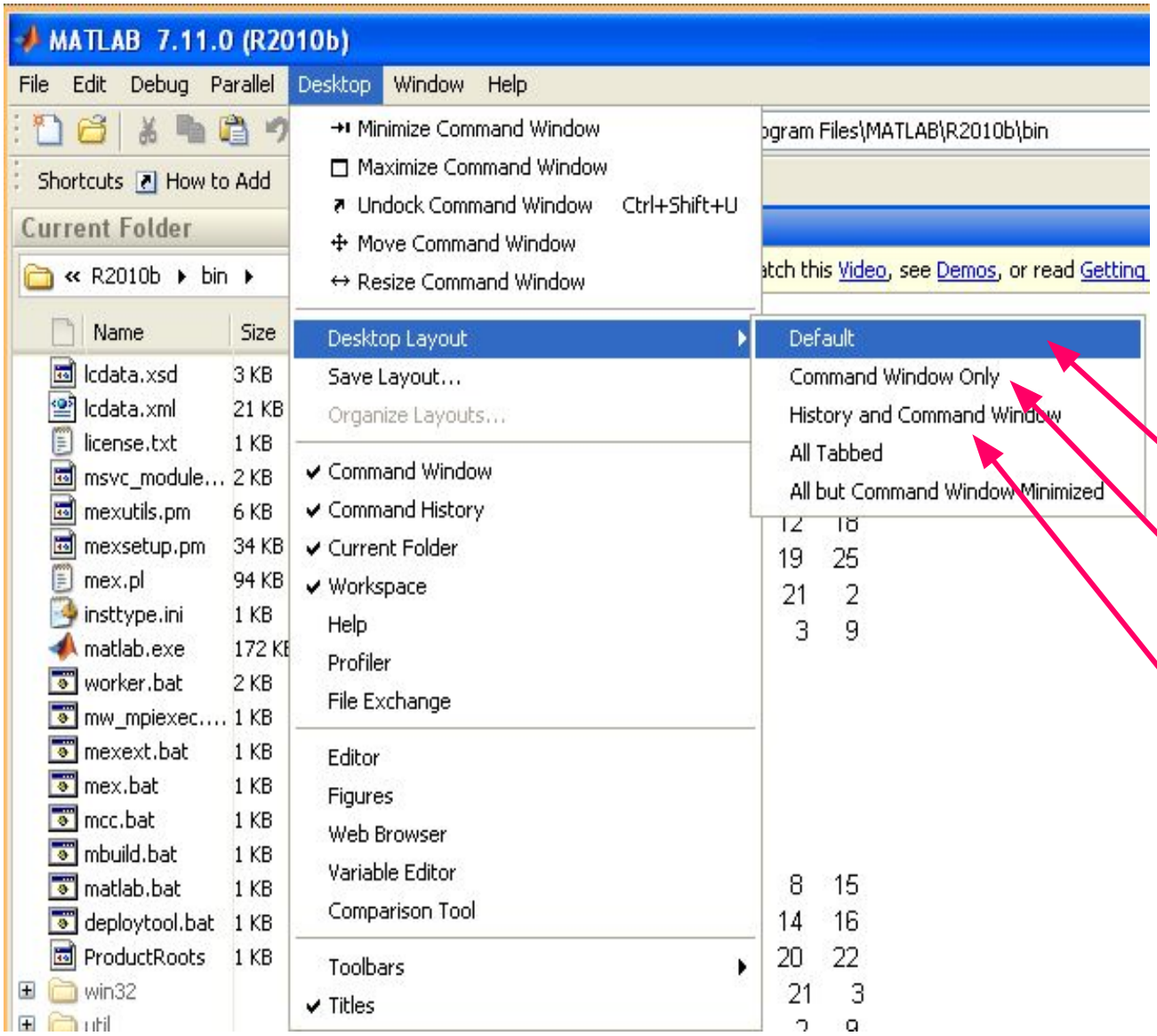


Главное меню



- Создать новый файл
- Открыть файл МАТЛАБ
- Сохранить рабочую область как файл типа .mat
- Предпочтения для интерфейсов МАТЛАБ (установка шрифтов, цветов и много другого)

Настройка вида рабочего стола



Меню Desktop

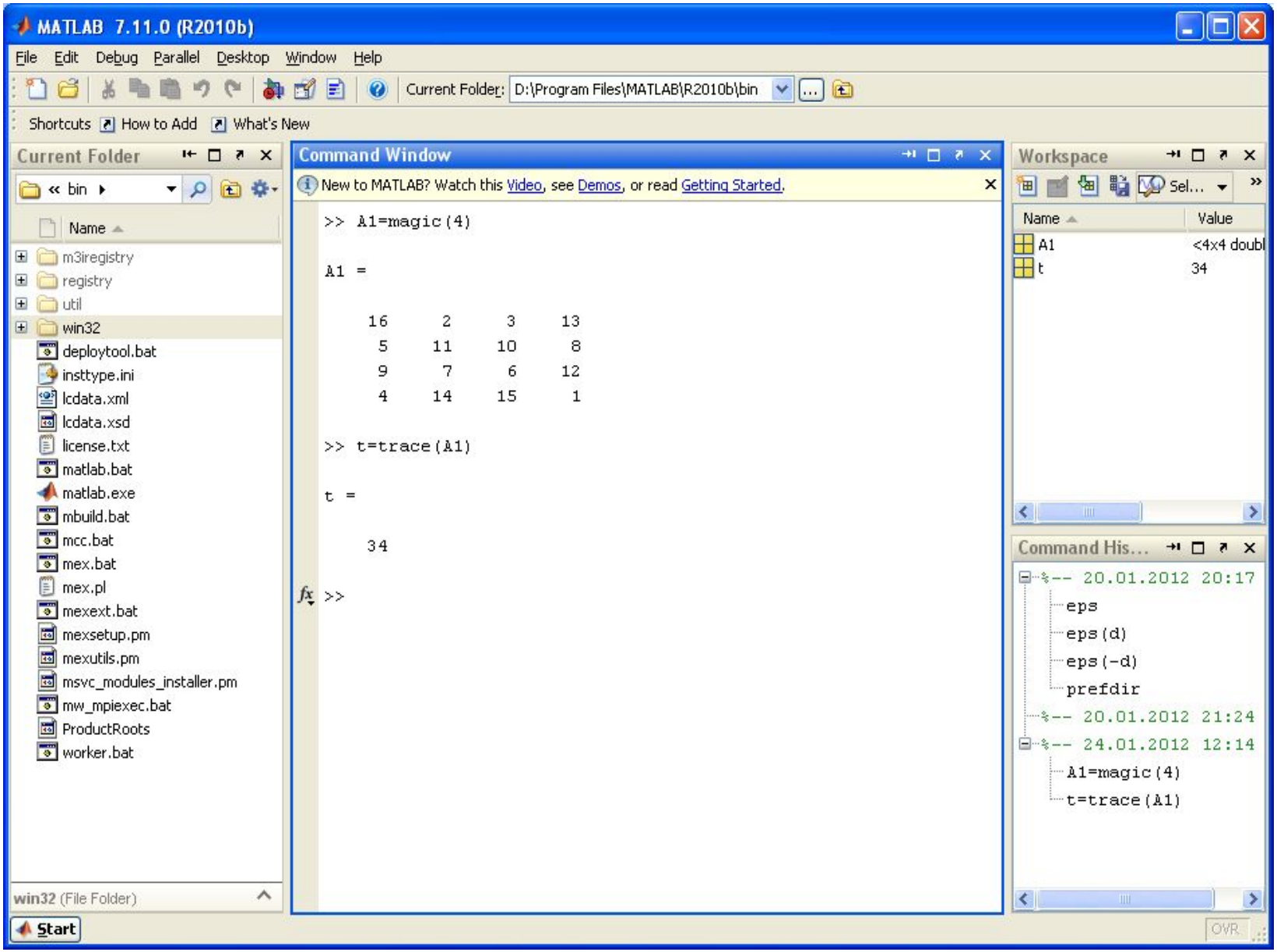
Команды управления схемой рабочего стола, задаётся количество и расположение окон.

По умолчанию

Только окно команд

Окно команд и история

По умолчанию



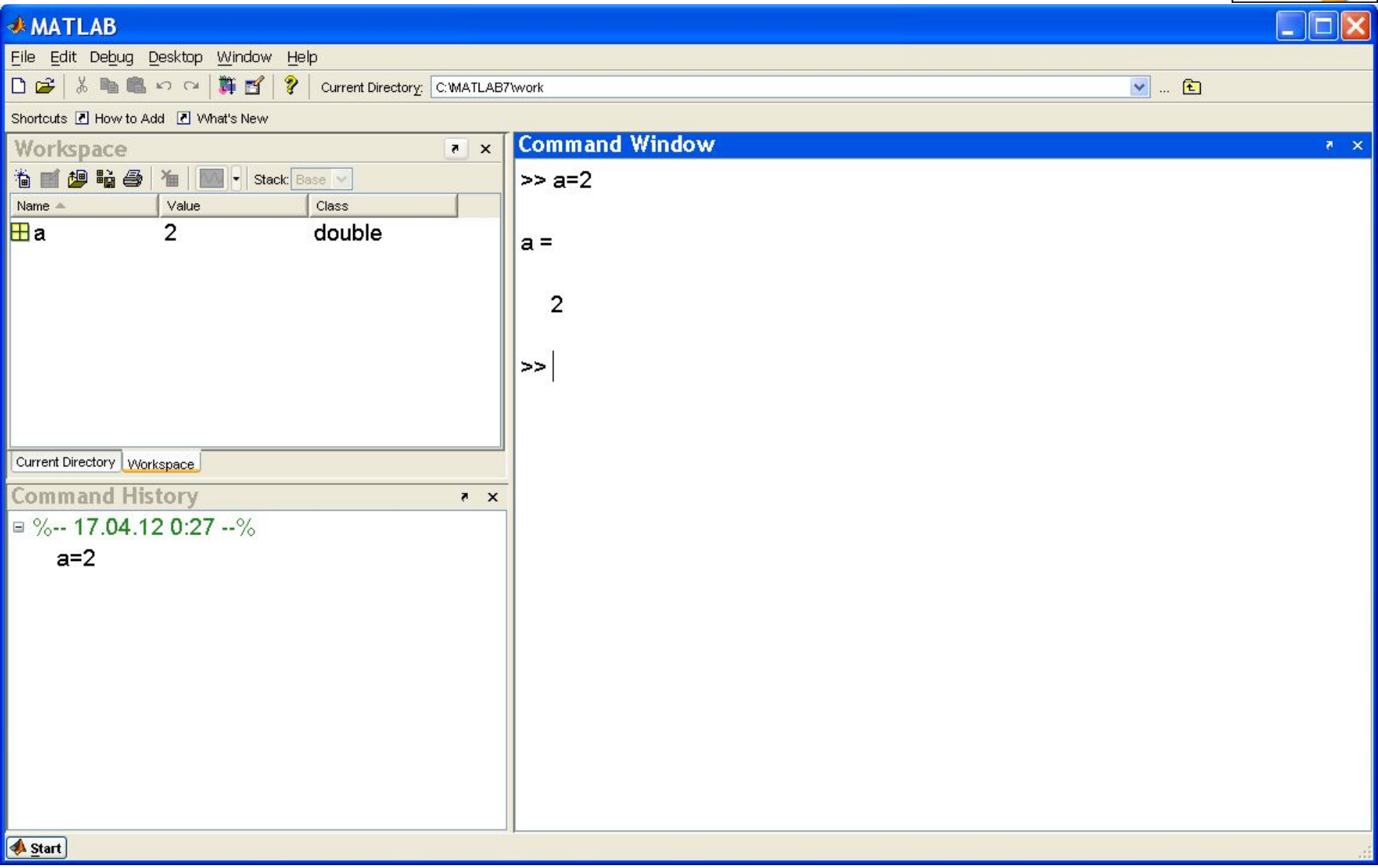
Работа в окне команд (режим калькулятора)

В системе МАТЛАБ можно

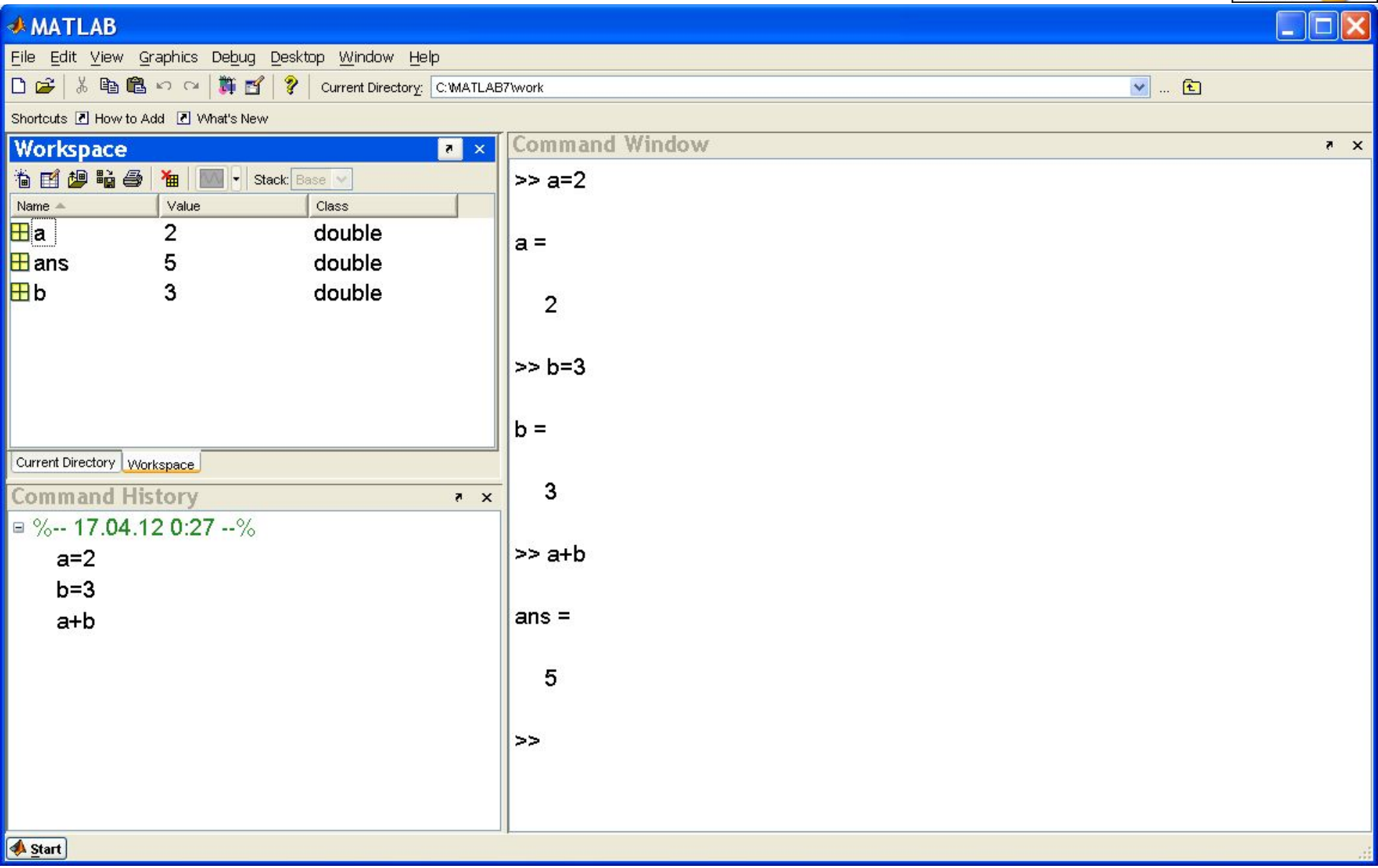
- **производить арифметические операции с**
  - ✔ **действительными и комплексными числами,**
  - ✔ **векторами и матрицами,**
- **вычислять функции,**
- **работать с полиномами и рядами,**
- **строить графики различных функций**

причём, непосредственно в интерактивном режиме, т.е. без подготовки программы

Работа в окне команд



Работа в окне команд



Понятие М-файла

Способы повторного ввода команд:

1. Использовать окно **Command History**
2. Применить **m-файл**

m-файл может содержать команды, а также управляющие структуры языка **MatLab.**

Вызов такого файла осуществляется заданием его имени.

Имя этого файла должно иметь расширение **m.**

Это текстовый файл – можно создавать и редактировать в любом текстовом редакторе (предпочтительнее – во встроенном редакторе **MatLab**).

m-файлы подразделяются на 2 типа:

- сценарии (**script**)
- функции (**function**)

M-файл (сценарий)

Содержит серию команд, которые выполняются в **режиме интерпретации** построчно.

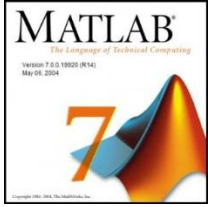
Если в команде имеется ошибка, она не обрабатывается, и система переходит в режим ожидания.

Сценарий работает **только** с переменными, расположенными в **рабочей области MatLab.**

M-функция

Отличие от сценария:

- Функция может компилироваться целиком с последующим размещением исполняемого кода в памяти
- Функция может иметь локальные переменные, размещаемые в собственной рабочей области
- В функции могут быть входные и выходные параметры



Команды управления окном

clc – очистки окна;

who – вывод имен активных переменных;

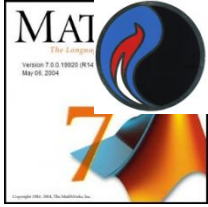
clear – удаление всех переменных;

clear a – удаление переменной a;

File->Save Workspace as… - сохранение в файле на диске содержимого рабочего пространства. Расширение файла **mat**.

File ->Load Workspace ->указание mat-файла для загрузки

Действительные и комплексные числа



-68

3.4567

7.13e13 – означает $7.13*10^{13}$

1.7977e+308 – максимальное число **realmax**

2.2251e-308 – минимальное число **realmin**

Inf для обозначения ∞

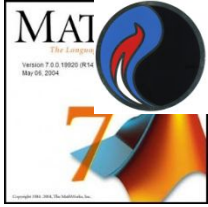
-Inf для обозначения **-∞**

NaN – не число (при делении 0/0)

2+3i

-6.789+0.834e-2*i

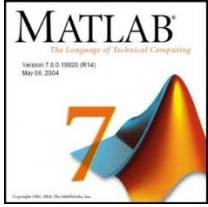
4-2j;



Форматы

- **format short** – 4 цифры после точки (по умолчанию)
- **format long** – 15 цифр после десятичной точки
- **format short e** – короткое с плавающей точкой
- **format long e** – длинное с плавающей точкой
- **format long g** –выбирается наиболее удачное
- **format short g** (с плавающей точкой или с фиксированной)
- **format rat** – формат для вывода рациональных чисел
- **format bank** – денежный формат (2 цифры после точки)

- **format loose** – обычный стиль вывода в окне команд
- **format compact** – компактный стиль вывода данных



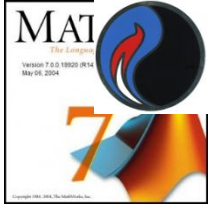
Вывод комментария на экран

disp (‘Результаты расчета')

Вывод значения a с точностью до 3 значащих цифр

vpa(a,3)

Знак точка с запятой в конце ввода предотвращает вывод результата на экран;



Арифметические операторы

Основные: **+ - * / ^**

Обратное деление **** - справа налево

Поэлементные: **.* ./ .^ .**

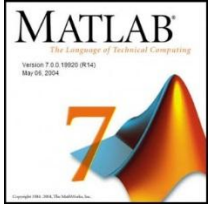
Операторы отношения

< > >= <= == ~=

Для комплексных чисел сравниваются только действительные части

Логические операторы

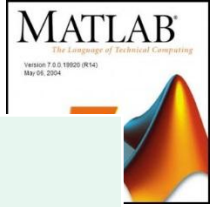
& — **И** **|** — **ИЛИ** **~** — **НЕ**



В математических выражениях операторы имеют определенный **приоритет исполнения**.

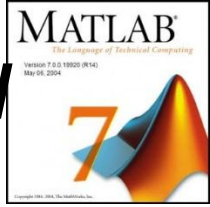
Например, в MATLAB приоритет логических операторов выше, чем арифметических, приоритет возведения в степень выше приоритетов умножения и деления, приоритет умножения и деления выше приоритета сложения и вычитания.

Элементарные алгебраические функции



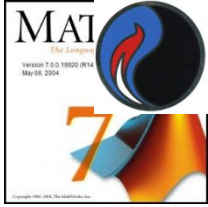
Функция	Описание
abs(x),	Вычисление абсолютного значения действительного числа x.
sqrt(x)	Вычисление квадратного корня
round(x)	Округление до целого.
fix(x)	Округление до ближайшего целого в сторону нуля.
floor(x)	Округляет до меньшего целого
ceil(x)	Округляет до большого целого
rem(x, y)	Вычисление остатка от деления x на y.
exp(x)	Вычисление e в степени x.
log(x)	Вычисление натурального логарифма числа x.
log10(x)	Вычисление десятичного логарифма числа x.

Тригонометрические функции



Функция	Описание
sin(x)	Вычисление синуса
cos(x)	Вычисление косинуса
tan(x)	Вычисление тангенса
asin(x)	Вычисление арксинуса
acos(x)	Вычисление арккосинуса
atan(x)	Вычисление арктангенса
atan2(y, x)	Вычисление арктангенса по координатам точки

Переменные



Результат вычислений присваивается переменной

```
>> x=2-3^2
x =
    -7
>> x1=5*x
x1 =
   -35
>> 1+1/2*4
ans =
     3
```

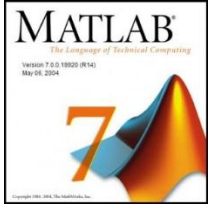
ans – имя переменной по умолчанию

Имя переменной – любая последовательность латинских букв и цифр, начинающаяся с буквы

В системе есть зарезервированные имена:
i, j, pi, имена стандартных функций и пр.

```
>> a=2;
>>
```

точка с запятой в конце строки отменяет вывод результатов

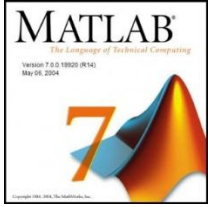


Основной объект в системе Matlab — это матрицы, или массивы. Даже скалярные величины, рассматриваются системой как матрицы 1 × 1.

Вектор (одномерный массив) представляет собой строку, т. е. матрицу размера 1×n, или столбец, т. е. матрицу размера m×1.

MatLab различает строчные и прописные буквы.

Количество воспринимаемых в MatLab символов в имени переменной составляет 31.

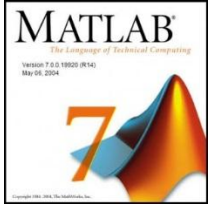


Чтобы задать вектор, достаточно перечислить его элементы, заключая их в квадратные скобки.

Элементы векторов-строк разделяются символами «,» (запятая) или « » (пробел).

Элементы векторов-столбцов разделяются символом «;» (точка с запятой) или символом перехода на новую строку.

Одномерные массивы



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

Двумерные массивы

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

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

a =

     1     2
     3     4
     5     6
```

- Доступ к элементу:

```
>> a(3,1)

ans =

     5

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

Диапазоны

- Функция **magic(n)** задает магическую матрицу **nxn** все ее элементы не превышают $\mathbf{n^2}$
- Можно использовать как для задания значений векторов, так и для задания диапазонов индексации

```
>> 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
```

Формирование матрицы A (3×3) с использованием операторов For и If

$$a_{i,j} = \begin{cases} 0.893 + e^{2j}, & \text{если} \quad j > i - 1 \\ 2.914 - \sin\left(\dfrac{j * \pi}{i} - i\right), & \text{в остальных случаях} \end{cases}$$

```
for i=1:3
    for j=1:3
        if j>i - 1
            a(i,j)=0.893+exp(2*j);
        else a(i,j)=2.914-sin(j*pi/i-i);
        end
    end
end
```

Создание массивов со случайными элементами

rand(n) и **rand(m,n)** генерируют матрицу **(nxn)** или **(mxn)** с элементами, распределенными по равномерному закону в промежутке (0,1)

randi([a,b],n,m) генерируют матрицу **(n x m)** с элементами в промежутке **(a,b)**

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
```

Удаление строк и столбцов

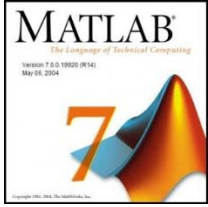
```
>> a = magic(5)

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
```



Операции над матрицами

a+b сложение скаляров, векторов или матриц

a−b вычитание скаляров, векторов или матриц

a*b умножение скаляров; матричное умножение

a.*b покомпонентное умножение элементов матриц

a^b возведение скаляра или матрицы в степень

a.^b возведение каждого элемента матрицы в степень

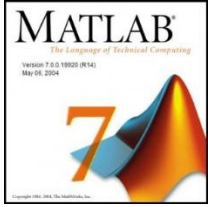
a/b деление скаляров;правое деление матриц, $\mathbf{a} \cdot \mathbf{b}^{-1}$

a./b покомпонентное деление элементов матриц

a\b левое деление матриц, т. е. $\mathbf{a}^{-1} \cdot \mathbf{b}$

A’ транспонирование матрицы

Операции над матрицами



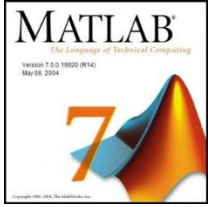
Функция **length(V)** рассчитывает количество элементов в векторе V.

Функция **max(V)** выдает значение максимального по значению элемента вектора V.

Функция **min(**V) извлекает минимальный элемент вектора V.

Функции **mean(V)** и **std(V)** определяют, соответственно, среднее значение и среднеквадратическое отклонение вектора V.

Операции над матрицами



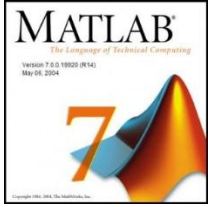
Функция сортировки **sort(V)** формирует вектор, элементы которого распределены в порядке возрастания их значений.

Функция **sum(V**) вычисляет сумму элементов вектора V.

Функция **prod(**V) выдает произведение всех элементов вектора V.

Функция **cumsum(V**) формирует вектор того же типа и размера, любой элемент которого является суммой всех предыдущих элементов вектора V (вектор кумулятивной суммы).

Операции над матрицами



abs(A) - модуль

det(A) - определитель матрицы

inv(A) - обратная матрица

diag(A) - главная диагональ матрицы

sum(A) - сумма по столбцам **(sum(A,1))**

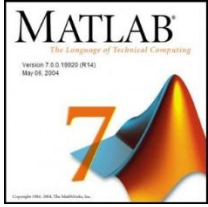
sum(A,2) - сумма по строкам

sum(diag(A)) - след матрицы

trace(A) - след матрицы

S=sum(sum(A)) - сумма матрицы

Операции над матрицами

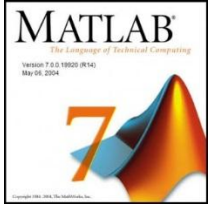


prod(A,1) **-** произведение элементов массива в столбцах (по умолчанию **prod(A)**)

prod(A,2) **-** произведение элементов массива в строках

sum(A') **-** сумма столбцов транспонированной матрицы

Операции над матрицами



sort(A) **-** сортировка по столбцам по возрастанию

sort(A,2) **-**сортировка по строкам по возрастанию

sort(A,'descend') **-** сортировка по столбцам по убыванию

-sort(-A,2) **-** сортировка по строкам по убыванию

Операции над матрицами

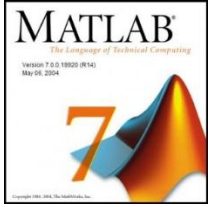
[b2,INDEX]=sort(b) - возвращает отсортированный массив и массив индексов элементов в исходном массиве

size(A) **-** размерность матрицы

max(A) - возвращает наибольший элемент, если A – вектор, или возвращает вектор-строку, содержащую максимальные элементы каждого столбца, если A -матрица

max(A,[ ],n) **-** возвращает наибольший элемент по столбцам при n=1, по строкам при n=2

min(A,[ ],n) **-** возвращает наименьший элемент по столбцам при n=1, по строкам при n=2



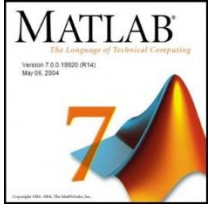
Дневник работы

Команда ***diary*** <имя файла>

открывает дневник, т.е. указывает системе, что все, что появится после этой команды на экране до следующей команды ***diary*** будет записано в упомянутый текстовый файл.

Прерывает запись в дневник команда открытия нового дневника или команда

diary off

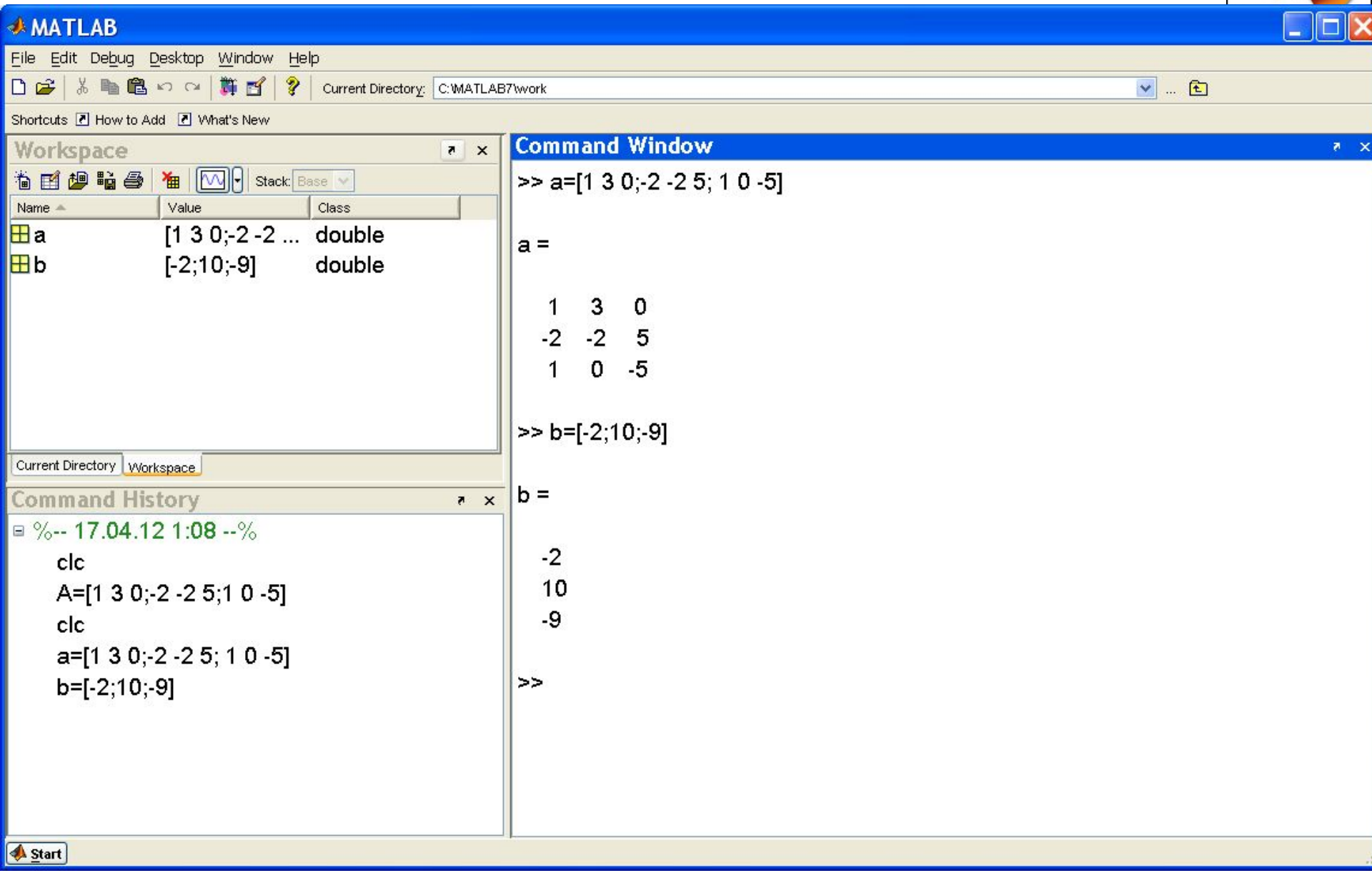


Решение системы линейных уравнений.

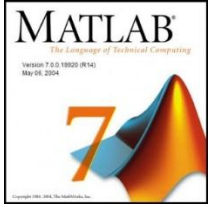
В матричном виде система имеет вид

$Ax = b$,

A , b , x – матрицы из коэффициентов при незвестных и вектор-столбцы, составленные соответственно из свободных членов и из неизвестных.

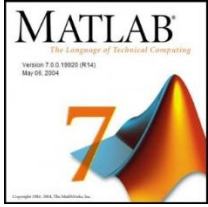

MATLAB
File Edit Debug Desktop Window Help
Current Directory: C:\MATLAB7\work
Shortcuts How to Add What's New
Workspace
Stack: Base
Name Value Class
a [1 3 0;-2 -2 ... double
b [-2;10;-9] double
Current Directory Workspace
Command History
%-- 17.04.12 1:08 --%
clc
A=[1 3 0;-2 -2 5;1 0 -5]
clc
a=[1 3 0;-2 -2 5; 1 0 -5]
b=[-2;10;-9]
Command Window
>> a=[1 3 0;-2 -2 5; 1 0 -5]
a =
1 3 0
-2 -2 5
1 0 -5
>> b=[-2;10;-9]
b =
-2
10
-9
>>
Start

Решение системы линейных уравнений




```
>> d=det(a)
d =
   -5
>> x=A\b
x =
    1
   -1
    2
```

Решение системы линейных уравнений

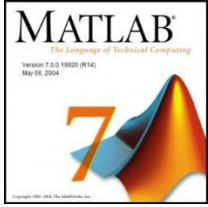


Решение x1=1, x2=-1, x3=2 легко проверить подстановкой в систему уравнений:

```
>> disp(A*x)
   -2.0000
   10.0000
   -9.0000
```

В результате получен вектор-столбец свободных членов. Система решена верно.

Решение системы линейных уравнений



Найдем обратную матрицу, а затем решение системы с помощью обратной матрицы:

```
>> A1=inv(A)
A1 =
   -2.0000   -3.0000   -3.0000
    1.0000    1.0000    1.0000
   -0.4000   -0.6000   -0.8000
>> A1*b
ans =
    1.0000
   -1.0000
    2.0000
```