



Учебный год 2017-2018

## Оглавление

Введение .....	5
Лабораторная работа 1. Основы MATLAB. Запись арифметических выражений... 7	
Запуск и элементарные операции .....	7
Выражения .....	8
Переменные.....	8
Числа.....	8
Операторы .....	9
Функции.....	9
Задание к лабораторной работе.....	11
Лабораторная работа 2. Скрипты и форматный вывод. ....	16
Функция fprintf.....	16
Задание к лабораторной работе.....	17
Лабораторная работа 3. Условные операторы. ....	28
Условный оператор if.....	28
Условные операторы switch и case .....	29
Задание к лабораторной работе.....	30
Лабораторная работа 4. Операторы цикла с предварительным условием и вывод в текстовый файл. ....	34
Цикл for.....	34
Цикл while .....	34
Оператор break .....	35
Функции save и load .....	35
Функции fwrite и fread.....	36
Задание к лабораторной работе.....	40
Файлы Excel .....	44
Файлы изображений .....	47
Лабораторная работа 5. Циклы с параметром и работа с матрицами.....	48
Создание матриц.....	48
Объединение и изменение матриц.....	49
Логическая индексация и функция find .....	50
Задание к лабораторной работе.....	52
Массивы ячеек .....	56
Символы и текст .....	56
Структуры .....	58
Лабораторная работа 6. Графики функций одной переменной. ....	60
Создание графика .....	60
Окна изображений .....	61
Добавление кривых на существующий график.....	62
Подграфики .....	63

Мнимые и комплексные данные .....	63
Управление осями .....	64
Подписи к осям, заголовки и легенда .....	65
Задание к лабораторной работе .....	67
Изображения .....	72
Печать и сохранение графики .....	72
Лабораторная работа 7. Многомерные вычисления, 3d-графика .....	73
Многомерные массивы .....	73
Функции mesh и surface .....	74
Визуализация функций двух переменных .....	74
Задание к лабораторной работе .....	76
Лабораторная работа 8. Программирование функций. ....	78
Функции .....	78
Глобальные переменные .....	79
Командно-функциональная двойственность .....	80
Функция eval .....	80
Векторизация .....	81
Предварительное определение .....	81
Задание к лабораторной работе .....	82
Лабораторная работа 9. Решение уравнений в MATLAB .....	86
Решение нелинейных уравнений .....	86
Нахождение экстремумов .....	87
Задание к лабораторной работе .....	89
Лабораторная работа 10. Графические интерфейсы, формы, функции обработки событий .....	92
Графические объекты .....	92
Управление объектами .....	92
Функции создания объектов .....	93
Свойства объекта .....	94
Функции set и get .....	95
Графический Пользовательский Интерфейс (GUI) .....	97
Редактор GUIDE .....	97
Задание к лабораторной работе .....	100
Анимация .....	102
Movie .....	103
Лабораторная работа 11. Математическая статистика .....	104
Законы распределения случайной величины в MATLAB .....	104
Функции для расчета плотности вероятности .....	105
Функции MATLAB для расчета функций распределения случайной величины .....	105
Функции вычисления статистических параметров .....	106
Гистограмма .....	107
Задание к лабораторной работе .....	108

Лабораторная работа 12. Основы работы в SIMULINK.....	110
Запуск пакета SIMULINK.....	110
Окно для создания модели.....	111
Построение модели .....	112
Основные разделы библиотеки SIMULINK.....	114
Генератор синусоидального сигнала Sine Wave.....	114
Генератор ступенчатого сигнала Step.....	116
Генератор импульсного сигнала Pulse Generator.....	117
Осциллограф Scope .....	119
Задание к лабораторной работе.....	121
Второе задание к лабораторной работе.....	121
Лабораторная работа 13. Моделирование электронных схем в SIMULINK. ....	123
Основные элементы библиотеки Simscape .....	123
Установка параметров моделирования .....	125
Задание к лабораторной работе.....	129
Пример m-файла для создания отчета.....	131

## **Введение**

Пакет MATLAB широко используется во всем мире при решении задач, связанных с матричными вычислениями. Название пакета образовано путем сокращения от MATrix LABoratory (матричная лаборатория). Операции и команды в MATLAB достаточно естественны и аналогичны математической записи формул на бумаге. MATLAB создавался как пакет программ, реализующих наиболее эффективные вычислительные алгоритмы линейной алгебры. Он организован таким образом, чтобы пользователь имел возможность применять при работе обычный математический язык.

В настоящее время пакет MATLAB представляет собой развитую интегральную программную среду, включающую собственный язык программирования. Он дает пользователю возможность быстро выполнять различные операции над векторами и матрицами, такие как умножение и обращение матриц, вычисление определителей, нахождение собственных чисел и векторов. Кроме того, в MATLAB входят операции вычисления обычных функций (алгебраических, тригонометрических, логических), решения алгебраических и дифференциальных уравнений, операции построения графиков и ряд других возможностей.

MATLAB является языком высокого уровня. По отдельным его командам можно выполнять такие сложные операции, как нахождение корней полиномов, решение линейных и нелинейных алгебраических уравнений, моделирование линейных динамических систем. Указанные операции являются элементарными функциями MATLAB.

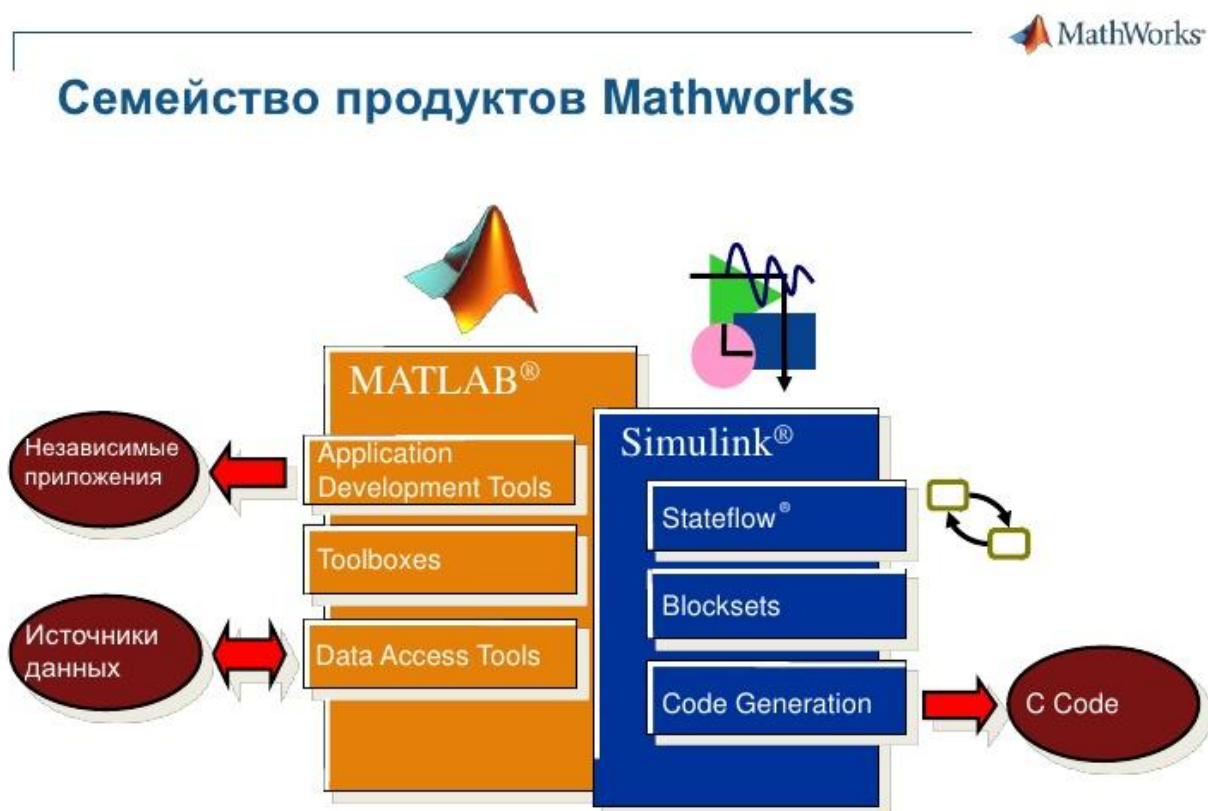
Помимо ядра, реализующего вычислительные алгоритмы общего назначения, в пакете MATLAB реализовано несколько десятков так называемых тулбоксов (библиотек специализированных подпрограмм), предназначенных для решения разнообразных практических задач. Например, тулбокс SYMBOLIC предназначен для выполнения символьных вычислений, а тулбокс CONTROL — для расчета и моделирования систем автоматического управления.

Типичное использование MATLAB — это:

- математические вычисления;
- создание алгоритмов;
- моделирование;
- анализ данных, исследования и визуализация;
- научная и инженерная графика;
- разработка приложений, включая создание графического интерфейса.

Вместе с пакетом MATLAB, поставляется также среда для визуального моделирования структурных схем SIMULINK, технология работы в которой в значительной степени копирует технику моделирования на аналоговых вычислительных машинах.

Общая структура системы MATLAB показана на рисунке ниже. Его левая часть соответствует ядру MATLAB, содержащему быстро выполняемые встроенные функции (сложение, умножение, тригонометрические и другие базовые функции) и так называемые m-функции, алгоритмы выполнения которых написаны на языке MATLAB. В центре показано семейство тулбоксов, каждый из которых содержит несколько десятков m-функций, справа — среда SIMULINK и связанные с ней средства (расширения и библиотеки блоков для разных приложений).



# **Лабораторная работа 1. Основы MATLAB. Запись арифметических выражений.**

## **Запуск и элементарные операции**

При запуске MATLAB, как правило, открывается интерфейс, который содержит меню, панель инструментов и два окна — командное окно (Command Window) и окно рабочего пространства (Workspace). Эти и другие окна можно активизировать во вкладке меню “Home” → “Layout”.

Команды вводятся в диалоговом режиме непосредственно в командное окно. Например, для того чтобы вычислить значение  $\sin 30^\circ$ , надо в командном окне набрать текст `sin(pi/6)` и нажать клавишу Enter. На экране появится ответ `ans=0.5`.

Разделительный знак «точка с запятой» ставится, чтобы не выводить на экран результаты промежуточных вычислений.

Когда выходная переменная не определена, MATLAB использует переменную `ans`, коротко от `answer` — ответ, для хранения результатов вычисления.

Совокупность переменных, созданных за время сеанса работы MATLAB, составляет рабочее пространство — `Workspace`. Это область памяти, доступная из командной строки MATLAB. Команды `who` и `whos`, показывают текущее содержание рабочего пространства. Команда `who` выдает краткий список, а команда `whos` размер и используемую память.

Вместо того чтобы набирать команды в окне MATLAB, их можно записать в текстовый файл с расширением `.m` (он называется m-файл, файл-сценарий или скрипт). Имя файла может быть любым, например, `vova.m`. Для того чтобы запустить скрипт, достаточно набрать его имя (без расширения) в командном окне

`vova`

Для вычисления только выделенной части скрипта удобно пользоваться горячей клавишей F9.

Файл должен находиться в рабочей директории MATLAB, либо путь к нему должен быть добавлен в список путей доступа. Для этого используются функции `path` и `addpath`.

Для оперативного получения справок об этих и других функциях используется команда `help`. Например, набрав в MATLAB

`help sind`

получим справку:

`sind` - Sine of argument in degrees

This MATLAB function returns the sine of the elements in X, which are expressed in degrees.

$Y = \text{sind}(X)$

Reference page for `sind`

See also `asin`, `asind`, `sin`

Больше информации о функциях и примерах их использования можно найти в Help Browser, который вызывается соответствующей кнопкой в меню “Home” или нажатием клавиши F1.

## **Выражения**

Как и большинство других языков программирования, MATLAB предоставляет возможность использования математических выражений, но в отличие от многих из них, эти выражения в MATLAB включают матрицы. Основные составляющие выражения:

- переменные;
- числа;
- операторы;
- функции.

## **Переменные**

В MATLAB нет необходимости в определении типа переменных или размерности. Когда MATLAB встречает новое имя переменной, он автоматически создает переменную и выделяет соответствующий объем памяти. Если переменная уже существует, MATLAB изменяет ее состав и если это необходимо выделяет дополнительную память. Например,

```
num_students = 25
```

создает матрицу 1x1 с именем `num_students` и сохраняет значение 25 в ее единственном элементе. Если после этого ввести

```
num_students = 'ten'
```

то `num_students` станет матрицей 1x3 с элементами типа `char`.

Имена переменных состоят из букв, цифр или символов подчеркивания. Длина имени переменной должна быть не более 31 символа. MATLAB чувствителен к регистрам, он различает заглавные и строчные буквы. Поэтому `A` и `a` — не одна и та же переменная. Чтобы увидеть матрицу, связанную с переменной, просто введите название переменной.

## **Числа**

MATLAB использует десятичную систему счисления с необязательной десятичной точкой и знаками плюс-минус для чисел. Буква `e` применяется для



определения множителя степени десяти. Мнимые числа записываются с суффиксами *i* или *j*. Некоторые примеры правильных чисел приведены ниже.

3	-99	0.0001
9.6397238	1.60210e-20	6.02252e23
1i	-3.14159j	3e5i

По умолчанию используется тип `double`, который имеет наибольшую точность представления вещественного числа и является потому универсальным типом. Для изменения типа данных используются функции `single`, `int8`, `uint8`, `int16` и др.

### **Операторы**

Выражения используют обычные арифметические операции и правила старшинства.

+	сложение
–	вычитание
*	умножение
/	деление
\	левое деление
^	степень
'	комплексно сопряженное транспонирование
()	определение порядка вычисления

### **Функции**

MATLAB предоставляет большое количество элементарных математических функций, таких как `abs`, `sqrt`, `exp`, `sin`. Вычисление квадратного корня или логарифма отрицательного числа не является ошибкой: в этом случае результатом является соответствующее комплексное число. MATLAB также предоставляет и более сложные функции, включая Гамма функцию и функции Бесселя. Большинство из этих функций имеют комплексные аргументы. Чтобы вывести список всех элементарных математических функций, наберите

```
help elfun
```

Для вывода более сложных математических и матричных функций, наберите

```
help specfun
```

```
help elmat
```

соответственно.

Некоторые функции, такие как `sqrt` и `sin`, — встроенные. Они являются частью MATLAB, поэтому они очень эффективны, но их вычислительные детали трудно доступны. В то время как другие функции, такие как `gamma` и `sinh`, реализованы в `m`-файлах. Поэтому можно легко посмотреть их код и, в случае необходимости, даже модифицировать его.

Несколько специальных функций предоставляют значения часто используемых констант.

<code>pi</code>	3.14159265...
<code>i</code>	мнимая единица, $\sqrt{-1}$
<code>j</code>	то же самое, что <code>i</code>
<code>eps</code>	относительная точность числа с плавающей точкой, $2^{-52}$
<code>realmin</code>	наименьшее число с плавающей точкой, $2^{-1022}$
<code>realmax</code>	наибольшее число с плавающей точкой, $(2-\epsilon)2^{1023}$
<code>Inf</code>	бесконечность
<code>NaN</code>	не число

Бесконечность появляется при делении на нуль или при выполнении математического выражения, приводящего к переполнению, т.е. к превышению `realmax`. Не число (`NaN`) генерируется при вычислении выражений типа `0/0` или `Inf - Inf`, которые не имеют определенного математического значения.

Имена функций не являются зарезервированными, поэтому возможно изменять их значения на новые, например

```
eps = 1.e-6
```

и далее использовать это значение в последующих вычислениях. Начальное значение может быть восстановлено следующим образом

```
clear eps
```

Список основных функций можно посмотреть в Help Browser → MATLAB → Mathematics.

(в MATLAB R2009 Help → MATLAB → Function Reference → Mathematics)

### Задание к лабораторной работе

1. В командном окне задать значения переменным.
2. Записать выражение на языке MATLAB. Если выражение не уместится в строке ввода, продолжить его в другой строке, используя символ продолжения (...).
3. Для вывода значения выражения не ставить после него точки с запятой.

Вариант	Задание
1	$a = -1,3; b = 0,91; c = 0,75; x = 2,32; k = 8$ $y = \sin \frac{a-x}{c} + 10^4 \sqrt[3]{\frac{a-kx^2}{2b}} + \frac{\cos kx^2}{\tan 3} - \frac{bc}{ax}$
	$k = 2; x = 0,32; d = 1,25; n = -4; b = 0,75; c = 2,2$ $z = 10^{-3} \tan kn - \frac{(x-d)(x^2+b^2)}{\sqrt[3]{x^2+b^2-cd}} - \frac{\cos kx}{\sin 5}$
2	$i = 5; k = -2; x = 0,1; a = 25,2; b = 2,35$ $y = \tan ik - \frac{ax^3 - b}{(a+b)^2} + 10^3 e^{-5} + \sqrt[3]{\frac{10^2  xk }{(a+b)^2}}$
	$a = -1,25; c = 0,05; d = 2,5; i = 5; x = 1,35$ $z = \frac{\sqrt{ c-d  + (a+c)^2}}{\sin 2i} + 10^{-3} e^{ix} - \frac{ c-d  + a^2}{\sqrt[3]{(a+c)^2}}$
3	$k = 2; x = -2,5; c = 0,31; a = 0,93; b = 5,61$ $y = \frac{\ln  kx }{\sin 7} - \sqrt{ x-a^2 } - \frac{10^4 a - b}{\cos kx} + \sqrt[3]{x-a^2} + c^3 x$
	$k = -2; a = 3,5; b = 0,35; x = 1,523$ $z = 10^4 \frac{ax}{b^2} - \left  \frac{a-b}{kx} \right  + \frac{\ln 3}{\sqrt[3]{ax+b^2}} - e^{-kx}$
4	$a = 1,7; b = -1,25; c = -0,3; x = 2,5; k = 3$ $y = \sqrt{\frac{abc}{2,4}} - \frac{0,7abc}{\sin 7} + 10^4 \sqrt[5]{ \cos kx } - \frac{ b-a }{kx}$

	$a = 1,3; b = 2,42; c = 0,83; x = 1,5; k = 2$ $z = \frac{ a^2 - b^2 }{\sin kx} - \frac{k^2 + \tan 3k}{e^{kx}} - 10^4 \sqrt[5]{ \sin kx - bc }$
5	$x = 0,29; a = -2,4; k = 3; c = 1,52$ $y = \frac{\sqrt[3]{\ln x + a^2}}{0,47x^2} - \left  0,47x^2 - \frac{10^4}{7} \cos^2 k \right  - \frac{c}{x}$
	$a = -2,5; b = 1,35; x = 2,75; i = 3; c = -0,72$ $z = \frac{1,5(a - b)^2}{ a - b c} + \frac{i}{5} + 10^3 \sqrt{ a - b } - \frac{2,5(a + x^2) \sin 7}{ix^2 + a^2 bc}$
6	$a = 3,5; i = 2; b = -0,7; x = 0,8$ $y = 10^4 \sin^2 i - \frac{0,32x^3 + 4x + b}{\cos ia} \sqrt[6]{0,32x^3 - b} +  b $
	$a = 4,72; b = 1,25; d = -0,01; x = 2,25; i = 2; k = 3$ $z = \frac{ax^2 +  d }{(a + b)^2} - 10^4 \sqrt[5]{\frac{kx}{(a + b)^2}} - \frac{\cos i}{\sin kx}$
7	$a = -3,25; x = 8,2; k = 4; b = 0,05; d = 0,95$ $y = (x - a) \cos k + \frac{\sqrt[5]{ x + a }}{2,4b} e^3 + 10^{-4} \frac{(x + a)^3 + x^4 d}{k(x - a)^3}$
	$x = 0,48; b = -0,31; c = 1,72; a = 2,01; k = 3$ $z = \sqrt[5]{ ax^2 - b^3 } + \ln kx - \frac{e^{kx} + c^2}{\sin kx} - 10^{-3} \sqrt{2157}$
8	$x = 2,5; b = 0,04; k = 3; n = 5$ $y = \frac{1}{9} + \frac{\sqrt{x^2 + b}}{0,4x} - 10^4 e^{kx} + \cos \sqrt{x^2 + b} + \frac{\sin 3}{(x^2 + b)n}$
	$x = 0,5; a = 2,71; c = 3,25; d = -3,53; k = 5$ $z = \frac{\sin(ax^2 - c)}{0,25k^2 xd} - \left  \sqrt[3]{x^2 + \ln 3} - \cos kx \right  + 10^4 x^5 cd$
9	$a = 0,02; x = -3,25; b = 2,5; c = 1,2; d = 0,5; k = 6$ $y = \frac{(ax - b)^2 +  d - b  - e^{kd}}{10^4 d^5 + b^2 + c} - \sin 2 + \sqrt[5]{d - b}$

	$a = -1,7; b = 2,32; c = 0,92; k = 2; x = 0,057$ $z = \sqrt{\left  \frac{\cos k^2 x - b}{a^2 + b^2} \right } - 10^4 e^7 + \frac{\tan k^2 x + \sqrt[3]{5}}{a - \sin k^2 x} - \frac{c}{k}$
10	$a = -1,52; b = -13,2; k = 2; n = 4; x = 1,4$ $y = 0,5 \frac{a^2 x +  b }{(a + b)^2 - b} + \frac{\sin k}{\cos nx} + 10^4 \sqrt[5]{a^2 x +  b }$
	$k = 3; a = -3,5; b = 0,35; n = 4; x = -0,02$ $z = \frac{abx + \tan 2k}{ a - b  + 0,5x} - 10^4 x \frac{\sin na}{\cos kx} - \frac{abx}{\sqrt[3]{a - b}}$
11	$a = -1,4; b = 25,3; x = 4,5; n = 4$ $y = 1,1 \frac{\sqrt[3]{(a + b)^2 +  \cos nx }}{\sin(a + b)} - e^2 + 10^{-3} \frac{n^2 x}{a + b}$
	$a = 2,75; b = 1,3; x = -7,85; d = 1,23; k = -2$ $z = 10^4 \frac{\sqrt{a^2 + b^2}}{x^2 + a^2} - 1,7 \frac{\sqrt{7}(a^2 + b^2)}{(a + b)kd} - \frac{\cos 2}{ x + d + k }$
12	$a = -5,1; x = 0,71; k = 4; b = 0,24$ $y = e^{ax} - \frac{\tan kx}{\sqrt{ a + x^2 }} - 10 \sin 2 + \frac{1}{3} - \frac{a + x^2}{kx} b$
	$a = 2,5; b = -5,25; x = 1,25; k = 4$ $z = \left(\frac{1}{3}\right)^3 e^{a+b} + \frac{\sqrt{15 - kx^2 - 0,41}}{10^{-2} a + b } + \frac{\ln(a + b)^2}{x + kx^2} - \sqrt{3}$
13	$d = 1,2; x = 0,75; c = 1,3; b = 2,35; i = 2; k = -3$ $y = \left(dx - \sqrt{\frac{ c - b }{x}}\right)^2 + 1,2 \tan i - 10^3 \frac{(c - b)^2}{dx} + \sqrt[3]{\cos kx}$
	$a = 1,2; k = 0,5; b = 0,1; x = 4,75$ $z = \sqrt[3]{(a^2 + x)x^2} - \frac{1}{\sqrt{\ln(b + x)}} + \sin\left(k + \frac{x^3}{a}\right)$
14	$a = 10; b = 5,43; c = 0,26; x = -0,55$

	$y = \frac{cx^2 + (abc)^3}{\cos cx} + \sqrt[4]{\frac{c+1}{x+b}} +  e^{cx-a} $
	$a = 3,5; b = 0,8; k = -2,3; x = -2,75$ $z = \frac{1}{7} - \cos(\sqrt{x^2 + b} + k) + \frac{e^{\frac{k}{x}} + \frac{a}{b}}{\sqrt[3]{308 + k}} + \frac{ a-b }{\tan \frac{k}{a}}$
15	$a = 6,52; b = 3,52; k = 1,6; x = 1$ $y = \sqrt[3]{b-kx} - \frac{a(a^2 + b)}{e^{2x+a}} + \left  \frac{\cos k^2 x}{a^3 + 2b^3} \right $
	$a = 3,27; b = 0,89; i = 0,5; x = -1,5$ $z = \frac{\sqrt{17 x }}{ae^{bx}} - \left(\frac{xi}{9}\right)^5 e^{a+b} + \frac{\ln(a+b)}{ix^2} \tan i$
16	$a = -1,5; b = 21,4; n = 3; x = 4,6$ $y = e^2 + 2,4 \frac{\sqrt[3]{(a-b)^2 +  \sin nx }}{\sin(a+b)} - 10^{-2} \frac{n^2 x}{a-b}$
	$a = -2,6; b = 3,56; x = 2,81; c = -0,84; i = 4$ $z = 10^4 \sqrt{ b-a } + \frac{0,5(a+b)^2}{ a-b c} - \frac{3,5(a+x^2) \cos x}{ix^2 + b^2 c} - \frac{x}{3}$
17	$a = 0,97; k = 3; x = -1,5; c = 0,51; b = 7,32$ $y = \sqrt{ x-a^2 } - \frac{10^3 a - c}{\sin kx} + \frac{\ln kx }{\sin b} - \sqrt[3]{x-c^2} + c^3 x$
	$a = 0,02; b = 2,5; c = 1,2; d = 0,5; k = 6; x = -3,25$ $z = \sqrt[3]{d-b} + \frac{(ax-b)^2 +  d-b  - e^{kd}}{10^3 d^3 + b^2 + c} - \sin b$
18	$a = 1,3; b = -0,2; x = 3,11; k = 0,4$ $y = \sin\left(k + \frac{x^2}{a}\right) - \frac{1}{\sqrt{\ln(b+x)}} + \sqrt[5]{(a^2 + x)x^2}$

	$x = 2,3; a = 1,5; b = -1,35; c = -0,2; k = 4$ $z = 10^5 \sqrt[3]{ \cos kx } - \frac{ b-a }{kx} - \sqrt{\frac{abc}{2,3}} - \frac{0,2abc}{\sin x}$
19	$x = 0,51; a = -4,2; b = 0,33; k = 3$ $y = 4 \sin 5 - \frac{1}{3} + e^{ax} - \frac{\tan kx}{\sqrt{ a+x^2 }} + \frac{a-x^2}{kx} b$
	$x = 1,1; a = 1,2; b = 2,57; c = 0,93; k = 3$ $z = \frac{k^3 + \tan 5k}{e^{kx}} + \frac{ a^2 - b^2 }{\sin kx} - 10^5 \sqrt[3]{ \sin kx - bc }$
20	$a = 2,5; b = 0,45; x = 1,522; k = -3$ $y = \frac{\ln 5}{\sqrt[3]{ax+b^2}} - 10^4 \frac{ax}{b^2} - \left  \frac{a-b}{kx} \right  + e^{-kx}$
	$x = 0,37; a = -1,9; c = 1,13; k = 2$ $z = \left  0,21x^2 - \frac{10^3}{5} \cos^2 k \right  - \frac{\sqrt[3]{\ln x + a^2}}{0,567x^2} - \frac{c}{x}$

## Лабораторная работа 2. Скрипты и форматный вывод.

### Функция fprintf

Для форматного вывода в командную строку или в файл используется функция `fprintf`. Для создания строки нужного формата используется функция `sprintf`.

```
fprintf(fid, format, a,b,...);  
str = sprintf(format,a,b,...);
```

Здесь `fid` — указатель на файл (если `fid = 1` или отсутствует, то данные выводятся в командную строку); `format` — строка формата данных; `a,b,...` — переменные для вывода/записи. Строка `format` может состоять из произвольного текста, спецификаторов для вывода переменных и управляющих символов.

Например, для записи числа и строки можно воспользоваться следующей записью:

```
s = 'Hello';  
x = 18000; y = 10.2;  
fprintf('Test %6d, (%.1f): %s\n', x, y, s);
```

и будет выведено:

```
Test 18000, (10.2): Hello
```

Спецификатор `%6d` говорит о том, что целые числа должны иметь 6 значащих цифр, а спецификатор `%.1f` означает, что после запятой будет отображено только 1 цифра. Наконец, в форматной строке был использован управляющий символ `\n` — переход на новую строку, который необходим для формирования строк в выходном файле.

Полный список возможных спецификаторов приведен в таблице:

Спецификатор	Описание
<code>%d</code> или <code>%i</code>	целочисленные значения
<code>%u</code>	беззнаковые целые значения
<code>%o</code>	числа в восьмеричной системе счисления
<code>%x</code>	числа в шестнадцатеричной системе счисления
<code>%X</code>	аналогично <code>%x</code> , только прописные буквы
<code>%f</code>	вещественные значения
<code>%e</code>	числа в экспоненциальном виде
<code>%E</code>	аналогично <code>%e</code> , только прописная E
<code>%g</code>	сокращенная форма вещественных чисел
<code>%G</code>	аналогично <code>%g</code> , только прописные буквы
<code>%s</code>	строковые данные
<code>%c</code>	символьные данные



### Задание к лабораторной работе

1. Написать m-файл, в котором вводятся все необходимые данные, а затем осуществляется расчёт и вывод результатов в командное окно в том виде, как указано в варианте задания.
2. Вместо многоточий выводятся числовые данные с использованием функции форматного вывода.
3. Ввод и вывод угловых значений осуществить в градусах, при этом помнить, что функции в MATLAB работают с радианами.

Вариант	Задание
1	<p>Вычислить координаты точки, делящий отрезок АВ в отношении <math>n_1:n_2</math>, по формулам:</p> $x = \frac{x_a + kx_b}{1 + k}, \quad y = \frac{y_a + ky_b}{1 + k}, \quad k = \frac{n_1}{n_2}$ <p>Вывести информацию в виде:</p> <ol style="list-style-type: none"> <li>1) Вывести заголовок: INPUT DATA</li> <li>2) Под заголовком вывести значения исходных данных в виде: <math>A = ( \dots , \dots ), B = ( \dots , \dots ), n1 = \dots, n2 = \dots</math></li> <li>3) Ниже вывести строку из дефисов: -----</li> <li>4) Пропустить 2 строки</li> <li>5) Отступив слева 5 позиций, вывести ответ в виде: COORDINATES: <math>x = \dots, y = \dots</math></li> </ol>
2	<p>Вычислить силу тока <math>I</math> в цепи, состоящей из соединённых последовательно сопротивления <math>R</math>, индуктивности <math>L</math> и ёмкости <math>C</math>, при напряжении в цепи <math>E</math> и частоте <math>F</math>:</p> $I = \frac{E}{\sqrt{R^2 + \left(2\pi FL - \frac{1}{2\pi FC}\right)^2}}$ <p>Вывести информацию в виде:</p> <ol style="list-style-type: none"> <li>1) Отступив слева 5 позиций, вывести значения исходных данных в виде: INPUT DATA <math>R = \dots \text{ Ohm}, \quad L = \dots \text{ H}</math> <math>C = \dots \text{ F}, \quad E = \dots \text{ V}</math> <math>F = \dots \text{ Hz}</math></li> </ol>

	<p>2) Вывести строку из дефисов: -----</p> <p>3) Отступив слева 5 позиций, вывести: ANSWER</p> <p>4) Под этой строкой вывести ответ в виде: CURRENT = ... A</p>
3	<p>Расстояние от точки с координатами <math>(x_0, y_0, z_0)</math> до плоскости, заданной уравнением <math>Ax + By + Cz + D = 0</math>, определяется по формуле:</p> $d = \frac{ Ax_0 + By_0 + Cz_0 + D }{\sqrt{A^2 + B^2 + C^2}}.$ <p>Найти расстояние от данной точки до параллельных плоскостей, заданных уравнениями:</p> $Ax + By + Cz + D_1 = 0,$ $Ax + By + Cz + D_2 = 0.$ <p>Вывести информацию в виде:</p> <p>1) Отступить 4 позиции слева и напечатать заголовок: INPUT DATA</p> <p>2) Пропустить строку</p> <p>3) На следующей строке вывести значения исходных данных в виде:</p> <p>X0 = ...      Y0 = ...      Z0 = ... D1 = ...      D2 = ... A = ...      B = ...      C = ...</p> <p>4) Для подчеркивания вывести строку из дефисов: -----</p> <p>5) Пропустить 2 строки и вывести ответ в виде:</p> <p>Answer</p> <p>The distance to 1 plane = ... The distance to 2 plane = ...</p>
4	<p>По заданным радиусам оснований <math>R</math> и <math>r</math>, образующей <math>l</math> и высоте <math>H</math> вычислить площадь поверхности <math>S</math> и объем <math>V</math> усеченного конуса:</p> $S = \pi(R + r)l + \pi R^2 + \pi r^2, \quad V = \frac{\pi H(R^2 + r^2 + Rr)}{3}.$ <p>Вывести информацию в виде:</p> <p>1) Отступив слева 5 позиций, вывести слова:</p>

	<p>INPUT DATA</p> <p>2) Под этими словами вывести данные в виде</p> <p>R = ...,                      r = ...,</p> <p>l = ...,                      H = ...</p> <p>3) Для подчеркивания вывести строку из дефисов</p> <p>-----</p> <p>4) Пропустить строку и вывести значения промежуточных результатов <math>\pi r^2</math> и <math>\pi R^3</math> в виде:</p> <p>INTERMEDIATE RESULTS:</p> <p>Base areas = ..., ...</p> <p>5) Пропустить строку и вывести ответ в виде:</p> <p>ANSWER:                      S = ...,                      V = ...</p>
5	<p>Вычислить медианы треугольника со сторонами <math>a</math>, <math>b</math>, <math>c</math> по формулам:</p> $m_a = 0,5\sqrt{2b^2 + 2c^2 - a^2}, \quad m_b = 0,5\sqrt{2a^2 + 2c^2 - b^2},$ $m_c = 0,5\sqrt{2a^2 + 2b^2 - c^2}.$ <p>Вывести информацию в виде:</p> <p>1) Вывести строку текста и наименования исходных данных:</p> <p>Sides of the triangle:</p> <p>a                                      b                                      c</p> <p>2) Под соответствующими наименованиями вывести значения</p> <p>3) Пропустить две строки</p> <p>4) Вывести ответ в виде:</p> <p>Medians of the triangle:</p> <p>ma = ...                      mb = ...                      mc = ...</p>
6	<p>Вычислить длину эллипса по приближённой формуле:</p> $l = 2\pi \left( \frac{3a + b}{2} - \frac{1}{2}\sqrt{ab} \right),$ <p>где <math>a</math> и <math>b</math> — полуоси эллипса.</p> <p>Вывести информацию в виде:</p> <p>1) Отступить 5 позиций и вывести ответ в виде:</p> <p>Length of ellipse = ...</p> <p>2) На следующей строке, отступив 3 позиций слева, вывести исходные данные в виде:</p> <p>Semiaxes of the ellipse</p> <p>a = ...,                      b = ...</p>

	<p>3) Пропустить 1 строку и вывести: #####</p> <p>4) Отступить 5 позиций и вывести слова: End of Task</p> <p>5) Далее вывести строку: #####</p>
7	<p>Вычислить высоты треугольника со сторонами <math>x</math>, <math>y</math>, <math>z</math> по формулам:</p> $h_x = \frac{2\sqrt{p(p-x)(p-y)(p-z)}}{x},$ $h_y = \frac{2\sqrt{p(p-x)(p-y)(p-z)}}{y},$ $h_z = \frac{2\sqrt{p(p-x)(p-y)(p-z)}}{z}.$ <p>1) Вывести значения исходных данных: Sides of the triangle: <math>x = \dots</math>, <math>y = \dots</math>, <math>z = \dots</math></p> <p>2) Вывести строку из дефисов: -----</p> <p>3) Пропустить 2 строки</p> <p>4) Отступить слева 5 позиций и вывести слова: Altitudes of the triangle:</p> <p>5) Под заголовком вывести ответ: <math>h_x = \dots</math> <math>h_y = \dots</math> <math>h_z = \dots</math></p> <p>6) Для подчеркивания вывести строку из дефисов -----</p>
8	<p>По двум сторонам <math>a</math>, <math>b</math> треугольника и углу между ними <math>C</math> найти третью сторону <math>c</math>, два других угла <math>A</math>, <math>B</math> и площадь треугольника <math>S</math>:</p> $c = a^2 + b^2 - 2ab \cos C, \quad A = \arccos \frac{b^2 + c^2 - a^2}{2bc},$ $B = \pi - (A + C), \quad S = \frac{1}{2}ac \sin B.$ <p>Выдать информацию в виде:</p> <p>1) Отступить 5 позиций и вывести слова:</p>

	<p>Sides of the triangle:</p> <p>2) Под словами вывести значения сторон:</p> <p><math>a = \dots</math></p> <p><math>b = \dots</math></p> <p><math>c = \dots</math></p> <p>3) Пропустить строку и, отступив 5 позиций, вывести слова:</p> <p>Angles of the triangle:</p> <p>4) Под словами вывести значения углов:</p> <p><math>A = \dots</math></p> <p><math>B = \dots</math></p> <p><math>C = \dots</math></p> <p>5) Пропустить строку и, отступив 5 позиций, вывести значение площади:</p> <p>Area of the triangle = ...</p>
9	<p>Вычислить координаты центра тяжести трёх материальных точек с массами <math>m_1, m_2, m_3</math> и координатами <math>(x_1, y_1), (x_2, y_2), (x_3, y_3)</math> по формулам:</p> $x_c = \frac{m_1 x_1 + m_2 x_2 + m_3 x_3}{m_1 + m_2 + m_3}, \quad y_c = \frac{m_1 y_1 + m_2 y_2 + m_3 y_3}{m_1 + m_2 + m_3}.$ <p>Выдать информацию в виде:</p> <p>1) Отступив слева 5 позиций и вывести слова:</p> <p>Input Data</p> <p>2) Для подчеркивания вывести строку из дефисов:</p> <p>-----</p> <p>3) Пропустить строку и вывести значения исходных данных:</p> <p><math>x1 = \dots, \quad x2 = \dots, \quad x3 = \dots,</math></p> <p><math>y1 = \dots, \quad y2 = \dots, \quad y3 = \dots,</math></p> <p><math>m1 = \dots, \quad m2 = \dots, \quad m3 = \dots</math></p> <p>4) Вывести строку из дефисов:</p> <p>-----</p> <p>5) пропустить 2 строки и вывести ответ:</p> <p><math>X_c = \dots,</math></p> <p><math>Y_c = \dots</math></p>
10	<p>По заданным сторонам треугольника <math>a, b, c</math> вычислить углы <math>A, B, C</math> и площадь <math>S</math> треугольника:</p> $A = \arccos \frac{b^2 + c^2 - a^2}{2bc}, \quad B = \arccos \frac{a^2 + c^2 - b^2}{2ac},$

	$C = \pi - (A + B), \quad S = \frac{1}{2}ab \sin C.$ <p>Выдать информацию в виде:</p> <p>1) Отступит 5 позиций и вывести слова: Answer of the task:</p> <p>2) Под словами вывести ответ: Angle A = ... Angle B = ... Angle C = ... Area S = ...</p> <p>3) Вывести строку из дефисов -----</p> <p>4) Пропустить 2 строки</p> <p>5) Отступив 3 пробела, вывести исходные данные: Input data: a = ..., b = ..., c = ...</p>
11	<p>По заданному радиусу <math>R</math> найти объём шара <math>V</math> и площадь сферы <math>S</math>:</p> $V = \frac{4}{3}\pi R^3, \quad S = 4\pi R^2.$ <p>Найти также объём шарового сегмента <math>V_c</math> высоты <math>H</math>:</p> $V_c = \pi H^2 \left( R - \frac{H}{3} \right).$ <p>Выдать на информацию в виде:</p> <p>1) Отступив 5 позиций, вывести заголовок: Input Data</p> <p>2) Вывести строку из дефисов: -----</p> <p>3) Пропустить строку и под заголовком вывести исходные данные: R = ..., H = ...</p> <p>4) Пропустить 2 строки</p> <p>5) Отступив слева 7 позиций, вывести ответ: Answer: V = ..., S = ..., Vc = ...</p>
12	<p>По двум углам треугольника <math>A</math> и <math>C</math> и стороне против одного из них <math>a</math> найти третий угол <math>B</math>, длины других сторон <math>b</math>, <math>c</math></p>

	<p>и площадь <math>S</math>:</p> $B = \pi - (A + C), \quad b = \frac{a \sin B}{\sin A}, \quad c = \frac{a \sin C}{\sin A},$ $S = \frac{1}{2} bc \sin A.$ <p>Выдать информацию в виде:</p> <p>1) Отступив 5 позиций, вывести слова: TRIANGLE ABC</p> <p>2) Вывести строку из дефисов: -----</p> <p>3) Под словами, пропустив одну строку, вывести значения сторон: SIDES:     <math>a = \dots</math>,     <math>b = \dots</math>,     <math>c = \dots</math></p> <p>4) Пропустив строку, под предыдущей записью вывести значения углов: ANGLES:    <math>A = \dots</math>,     <math>B = \dots</math>,     <math>C = \dots</math></p> <p>5) Через строку под последней записью вывести значение площади: AREA        <math>S = \dots</math></p>
13	<p>Расстояние от точки <math>N(x_0, y_0)</math> до прямой, заданной уравнением <math>y = a + bx</math>, вычисляется по формуле:</p> $d = \frac{ a + bx_0 + y_0 }{\sqrt{b^2 + 1}}$ <p>Найти расстояния <math>d_1</math> и <math>d_2</math> от точки <math>N</math> до прямых, заданных уравнениями:</p> $y = a_1 + bx$ $y = a_2 + bx$ <p>Выдать информацию в виде:</p> <p>1) Отступить 5 позиций и вывести значения исходных данных: POINT N:     <math>x_0 = \dots</math>,     <math>y_0 = \dots</math>               <math>a_1 = \dots</math>,     <math>a_2 = \dots</math>               <math>b = \dots</math></p> <p>2) Вывести строку из дефисов: -----</p> <p>3) Пропустить 2 строки и, отступив 7 позиций, вывести слова: ANSWER:</p> <p>4) Под словами напечатать ответ в виде: <math>d_1 = \dots</math>,</p>

	<p>d2 = ...</p> <p>5) Вывести строку из дефисов:</p> <p>-----</p>
14	<p>Найти объём конуса <math>V</math> и площадь его поверхности <math>S</math> по заданному радиусу основания <math>R</math>, высоте <math>H</math> и образующей <math>L</math>:</p> $V = \frac{1}{3}\pi R^2 H, \quad S = S_{side} + S_{base} = \pi RL + \pi R^2$ <p>Выдать информацию в виде:</p> <p>1) Отступить 5 позиций и вывести слова:</p> <p>The task is done</p> <p>2) Пропустить 2 строки</p> <p>3) Отступив 7 позиций вывести исходные данные:</p> <p>Input data</p> <p>R = ...,          H = ...,          L = ...</p> <p>4) Вывести строку из дефисов:</p> <p>-----</p> <p>5) Напечатать ответ:</p> <p>Answer:    V = ...</p> <p>              S = Sside+Sbase = ... + ... = ...</p>
15	<p>По заданным двум углам треугольника <math>B</math>, <math>C</math> и стороне между ними <math>a</math> найти третий угол <math>A</math>, длины других сторон <math>b</math>, <math>c</math> и площадь треугольника <math>S</math>:</p> $A = \pi - (B + C), \quad b = \frac{a \sin B}{\sin A}, \quad c = \frac{a \sin C}{\sin A}, \quad S = \frac{1}{2}ac \sin B$ <p>Выдать информацию в виде:</p> <p>1) Отступить 5 позиций слева и вывести слова:</p> <p>Input Data</p> <p>2) Под словами вывести значения исходных данных:</p> <p>Angle B = ...,          Angle C = ...</p> <p>Side a = ...</p> <p>3) Вывести строку из дефисов:</p> <p>-----</p> <p>4) Пропустить две строки и, отступив 7 позиций, вывести ответ:</p> <p>ANSWER: angle A = ...</p> <p>              side b = ...</p> <p>              side c = ...</p>



	area = ...
16	<p>Вычислить координаты точки, делящий отрезок АВ в отношении <math>n_1:n_2</math>, по формулам:</p> $x = \frac{x_a + kx_b}{1 + k}, \quad y = \frac{y_a + ky_b}{1 + k}, \quad k = \frac{n_1}{n_2}$ <p>Вывести информацию в виде:</p> <p>1) Отступить 5 позиций слева и вывести заголовок: Input Data:</p> <p>2) Под заголовком вывести значения исходных данных в виде: A = ( ... , ... ), B = ( ... , ... ), n1 = ..., n2 = ...</p> <p>3) Пропустить 2 строки</p> <p>4) Отступив слева 5 позиций, вывести ответ в виде: Answer: x = ..., y = ...</p> <p>5) Вывести строку из дефисов: -----</p>
17	<p>Вычислить силу тока <math>I</math> в цепи, состоящей из соединённых последовательно сопротивления <math>R</math>, индуктивности <math>L</math> и ёмкости <math>C</math>, при напряжении в цепи <math>E</math> и частоте <math>F</math>:</p> $I = \frac{E}{\sqrt{R^2 + \left(2\pi FL - \frac{1}{2\pi FC}\right)^2}}$ <p>Вывести информацию в виде:</p> <p>1) Вывести значения исходных данных в виде: Input Data: R = ... Ohm, L = ... H C = ... F, E = ... V F = ... Hz</p> <p>2) Вывести строку из дефисов: -----</p> <p>3) Отступив слева 7 позиций, вывести: Answer:</p> <p>4) Под этой строкой, отступив слева 5 позиций, вывести ответ в виде: Current = ... A</p> <p>5) Вывести строку: #####</p>

18	<p>Расстояние от точки с координатами <math>(x_0, y_0, z_0)</math> до плоскости, заданной уравнением <math>Ax + By + Cz + D = 0</math>, определяется по формуле:</p> $d = \frac{ Ax_0 + By_0 + Cz_0 + D }{\sqrt{A^2 + B^2 + C^2}}.$ <p>Найти расстояние от данной точки до параллельных плоскостей, заданных уравнениями:</p> $Ax + By + Cz + D_1 = 0,$ $Ax + By + Cz + D_2 = 0.$ <p>Вывести информацию в виде:</p> <p>1) Отступить 3 позиции слева и напечатать заголовок: Input Data:</p> <p>2) На следующей строке под заголовком вывести значения исходных данных в виде:</p> <p>X0 = ...      Y0 = ...      Z0 = ... D1 = ...      D2 = ... A = ...      B = ...      C = ...</p> <p>3) Пропустить 1 строку</p> <p>4) Для подчеркивания вывести строку из дефисов: -----</p> <p>5) Пропустить 1 строку и вывести ответ в виде:</p> <p style="text-align: center;">Answer:</p> <p>The distance to 1 plane = ... The distance to 2 plane = ...</p>
19	<p>По заданным радиусам оснований <math>R</math> и <math>r</math>, образующей <math>l</math> и высоте <math>H</math> вычислить площадь поверхности <math>S</math> и объём <math>V</math> усечённого конуса:</p> $S = \pi(R + r)l + \pi R^2 + \pi r^2, \quad V = \frac{\pi H(R^2 + r^2 + Rr)}{3}.$ <p>Вывести информацию в виде:</p> <p>1) Отступив слева 7 позиций, вывести слова: Input Data:</p> <p>2) Под этими словами вывести данные в виде:</p> <p>R = ...,      r = ..., l = ...,      H = ...</p> <p>3) Пропустить строку</p> <p>4) Для подчеркивания вывести строку из дефисов</p>

	<p>-----</p> <p>5) Вывести значения промежуточных результатов <math>\pi r^2</math> и <math>\pi R^3</math> в виде: Intermediate Results: Base areas = ..., ...</p> <p>6) Пропустить строку и вывести ответ в виде: Answer:            S = ...,            V = ...</p>
20	<p>Вычислить медианы треугольника со сторонами <math>a</math>, <math>b</math>, <math>c</math> по формулам:</p> $m_a = 0,5\sqrt{2b^2 + 2c^2 - a^2}, \quad m_b = 0,5\sqrt{2a^2 + 2c^2 - b^2},$ $m_c = 0,5\sqrt{2a^2 + 2b^2 - c^2}.$ <p>Вывести информацию в виде:</p> <p>1) Вывести строки текста и наименования исходных данных: Input Data Sides of the triangle: a                      b                      c</p> <p>2) Под соответствующими наименованиями вывести значения</p> <p>3) Пропустить строку и вывести ответ в виде: Medians of the triangle: ma = ...                      mb = ...                      mc = ...</p>

## **Лабораторная работа 3. Условные операторы.**

### **Условный оператор if**

Оператор `if` вычисляет логическое выражение и выполняет группу операторов, если выражение истинно. Необязательные ключевые слова `elseif` и `else` служат для выполнения альтернативных групп операторов. Ключевое слово `end`, которое согласуется с `if`, завершает последнюю группу операторов. Таким образом, все группы операторов заключены между четырьмя ключевыми словами, без использования фигурных или обычных скобок.

Алгоритм MATLAB для создания магического квадрата порядка  $n$  включает три разных случая:  $n$  нечетное,  $n$  четное, но не делится на 4, и  $n$  четное и делится на 4. Ниже приведен пример соответствующего кода.

```
if rem(n,2) ~= 0
    M = odd_magic(n)
elseif rem(n,4) ~= 0
    M = single_even_magic(n)
else
    M = double_even_magic(n)
end
```

В этом примере три случая являются взаимно исключающими, но если бы это было не так, то выполнялось бы первое истинное условие.

Важно понять, как оператор отношения и оператор `if` работают с матрицами. Когда вы хотите узнать, равны ли две переменные, нужно использовать следующую конструкцию

```
if (A == B), ...
```

Это правильный код MATLAB и он осуществляет то, что вы ожидаете, если  $A$  и  $B$  являются скалярами. Но когда  $A$  и  $B$  — матрицы,  $A == B$  не выполнится, если они не равны поэлементно. Фактически, если  $A$  и  $B$  имеют различные размеры, MATLAB выдаст ошибку.

Правильный способ определения равенства между двумя переменными — это использование функции `isequal`:

```
if isequal(A,B), ...
```

Далее приведен другой пример, который исследует этот вопрос. Если  $A$  и  $B$  являются скалярами, то нижеприведенная программа никогда не приведет к неожиданной ситуации. Но для большинства пар используемых матриц ни одно из условий  $A > B$ ,  $A < B$  или  $A == B$  не является истинным для всех элементов и поэтому выполняется случай `else`.

```
if A > B
    'greater'
elseif A < B
```

```

        'less'
elseif A == B
        'equal'
else
        error ( ' Непредвиденная ситуация ' )
end

```

Некоторые функции могут быть полезны для матричного сравнения при использовании с оператором if, например isequal, isempty, all, any.

### **Условные операторы switch и case**

Оператор switch выполняет группу операторов, базируясь на значении переменной или выражения. Ключевые слова case и otherwise разделяют эти группы. Выполняется только первый соответствующий случай. Необходимо использовать end для согласования с оператором switch.

Логика алгоритма магических квадратов может быть описана на следующем примере

```

switch (rem(n,4)==0) + (rem(n,2)==0)
    case 0
        M = odd_magic(n)
    case 1
        M = single_even_magic(n)
    case 2
        M = double_even_magic(n)
    otherwise
        error( ' Это невозможно ' )
end

```

Замечание для программистов Си. В отличие от языка Си, оператор switch в MATLAB не "проваливается". Если первый случай является истинным, другие случаи не выполняются. Таким образом, нет необходимости в использовании оператора break для каждого case.

### Задание к лабораторной работе

1. В скрипте ввести все необходимые данные.
2. Выполнить расчёт с использованием условных операторов и вывести исходные данные и результаты в командное окно.
3. В качестве исходных данных следует взять такие, чтобы можно было проверить вычисления по разным направлениям.

Вариант	Задание
1	Найти сумму положительных из четырёх заданных переменных.
	Определить номер квадранта на координатной плоскости, в котором находится точка с заданными координатами.
2	Найти все пары одинаковых значений среди четырёх переменных.
	Найти максимальное значение из четырёх заданных переменных и вывести её имя.
3	Найти значение функции распределения при заданном аргументе $x$ и коэффициенте $a$ : $F(x) = \begin{cases} 1, & x \geq a \\ \frac{1}{2} + \frac{1}{\pi} \arcsin \frac{x}{a}, & -a < x < a \\ 0, & x \leq -a \end{cases}$
	Заданы четыре переменные. Найти среди них пары значений, отличающихся друг от друга на две единицы.
4	Используя метод Крамера, решить систему уравнений с двумя неизвестными. Если система решения не имеет, то вывести сообщение об этом. $\begin{cases} a_{11}x_1 + a_{12}x_2 = a_{13} \\ a_{21}x_1 + a_{22}x_2 = a_{23} \end{cases}$
	Заданы четыре переменные. Наименьшую из них заменить на сумму остальных. Вывести её с указанием имени переменной.
5	Заданы четыре переменные. Переменные, отличные по величине от 3 и 7, заменить нулями.

	Заданы четыре переменные. Подсчитать количество отрицательных и количество нулевых из них.
6	Заданы четыре переменные. Известно, что три из них равны между собой, а одна — отлична от других. Вывести имя и значение этой переменной.
	Найти произведение отрицательных из четырёх заданных переменных.
7	Заданы две фигуры: квадрат задан длиной стороны, а круг — длиной радиуса. Определить, какая из них имеет большую площадь и больший периметр и во сколько раз.
	<p>Заданы аргументы <math>x_1, x_2, x_3</math> и соответствующие значения <math>y_1, y_2, y_3</math> функции <math>y(x)</math>. Вычислить значение функции в некоторой точке <math>x</math>, лежащей в интервале <math>x_1 \leq x \leq x_3</math>, используя формулу линейной интерполяции:</p> $y(x) = \begin{cases} y_1 + \frac{x - x_1}{x_2 - x_1} (y_2 - y_1), & x_1 \leq x \leq x_2 \\ y_2 + \frac{x - x_2}{x_3 - x_2} (y_3 - y_2), & x_2 \leq x \leq x_3 \end{cases}$
8	Ввести три переменные и вывести их в порядке убывания (вывести имена и значения переменных).
	Заданы четыре переменные. Подсчитать количество и произведение значений, попавших в интервал $[1; 5]$ .
9	Заданы четыре переменные. Все отрицательные из них заменить абсолютным значением (чтобы они стали положительными) и увеличить в 2 раза.
	Задан четырёхугольник координатами своих вершин. Определить, является ли он ромбом, параллелограммом.
10	Заданы четыре переменные, подсчитать количество равных нулю, положительных и отрицательных.
	На плоскости заданы четыре точки $M_1(x_1, y_1)$ , $M_2(x_2, y_2)$ , $M_3(x_3, y_3)$ , $M_4(x_4, y_4)$ . Определить, к какой из точек $M_1$ , $M_2$ или $M_3$ точка $M_4$ расположена ближе.

11	Вычислить вещественные корни уравнения $ax^2 + bx + c = 0$ при заданных коэффициентах. Если задача не имеет решения, то выдать строку: Уравнение не имеет действительных корней!
	Даны четыре переменные $a, b, c, d$ . Определить, какая из них делится без остатка на 3.
12	Задано четыре значения. Определить, какие из них целые.
	Даны четыре переменные $a, b, c, d$ . Найти среди них переменные, наиболее близкие по значению к числу $x$ .
13	Задать значения $x_1, x_2, x_3, x_4$ так, чтобы они располагались в порядке возрастания $x_1 < x_2 < x_3 < x_4$ . Ввести $x$ и определить номер группы, в которую это значение попадает: $x < x_1$ группа N1 $x_1 \leq x < x_2$ группа N2 $x_2 \leq x < x_3$ группа N3 $x_3 \leq x < x_4$ группа N4 $x \geq x_4$ группа N5
	Заданы четыре переменные. Подсчитать количество и сумму значений, не попавших в интервал $[-5; 5]$ .
14	Ввести три переменные и вывести их в порядке возрастания (вывести имена и значения переменных).
	Найти минимальное и максимальное значения из четырёх заданных переменных.
15	Определить, можно ли из отрезков с длинами $x, y, z$ построить треугольник. Проверить также, является ли треугольник равносторонним или равнобедренным.
	Задано четыре значения. Определить, какие из них дробные.
16	Найти сумму положительных из четырёх заданных переменных.
	Заданы четыре переменные. Переменные, отличные по величине от 2 и 5, заменить нулями.



17	Определить номер квадранта на координатной плоскости, в котором находится точка с заданными координатами.
	Заданы четыре переменные. Наименьшую из них заменить на среднее остальных. Вывести её с указанием имени переменной.
18	Заданы четыре переменные. Найти среди них пары значений, отличающихся друг от друга на три единицы.
	Даны четыре переменные $a, b, c, d$ . Определить, какая из них делится без остатка на 4.
19	Заданы четыре переменные. Подсчитать количество и сумму значений, не попавших в интервал $[-4; 6]$ .
	На плоскости заданы четыре точки $M_1(x_1, y_1)$ , $M_2(x_2, y_2)$ , $M_3(x_3, y_3)$ , $M_4(x_4, y_4)$ . Определить, к какой из точек $M_1$ , $M_2$ или $M_3$ точка $M_4$ расположена ближе.
20	Заданы четыре переменные. Известно, что три из них равны между собой, а одна — отлична от других. Вывести имя и значение этой переменной.
	Определить, можно ли из отрезков с длинами $x, y, z$ построить треугольник. Проверить также, является ли треугольник равносторонним или равнобедренным.

## **Лабораторная работа 4. Операторы цикла с предварительным условием и вывод в текстовый файл.**

### **Цикл for**

Цикл for повторяет группу операторов фиксированное, предопределенное число раз. Ключевое слово end очерчивает тело цикла.

```
for n = 3:10
    r(n) = rank(magic(n));
end
r
```

Точка с запятой после выражения в теле цикла предотвращает повторения вывода результатов на экран, а r после цикла выводит окончательный результат.

### **Цикл while**

Цикл while повторяет группу операторов определенное число раз, пока выполняется логическое условие. Ключевое слово end очерчивает используемые операторы.

Ниже приведена полная программа, иллюстрирующая работу операторов while, if, else и end, которая использует метод деления отрезка пополам для нахождения нулей полинома.

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if sign(fx) == sign(fa)
        a = x;
        fa = fx;
    else
        b = x;
        fb = fx;
    end
end
x
```

Результатом будет корень полинома  $x^3-2x-5$

```
x =
2.0946
```

Для оператора while верны те же предостережения относительно матричного сравнения, что и для оператора if, которые обсуждались ранее.

## **Оператор break**

Оператор `break` позволяет досрочно выходить из циклов `for` или `while`. Во вложенных циклах `break` осуществляет выход только из самого внутреннего цикла.

Ниже представлен улучшенный вариант предыдущего примера:

```
a = 0; fa = -Inf;
b = 3; fb = Inf;
while b-a > eps*b
    x = (a+b)/2;
    fx = x^3-2*x-5;
    if fx == 0
        break
    elseif sign(fx) == sign(fa)
        a = x;
        fa = fx;
    else
        b = x;
        fb = fx;
    end
end
x
```

## **Функции save и load**

В самом простом случае для сохранения и последующей загрузки каких-либо данных в MATLAB предусмотрены функции `save` и `load`.

Функция `save` позволяет сохранять произвольные переменные программы в файл, который будет (по умолчанию) располагаться в рабочем каталоге и иметь расширение `.mat`. Соответственно функция `load` позволяет загрузить из указанного `mat`-файла ранее сохраненные переменные. Ниже представлен пример использования данных функций:

```
x = ones(5);
y = 5;
s = 'hello';
save('params', 'x', 'y', 's');
```

```
x = zeros(5);
y = 0;
s = '';
```

```
load('params', 'x', 'y', 's');
disp(x);
disp(y);
```

```
disp(s);
```

В данной программе сначала выполняется инициализация переменных *x*, *y*, *s*. Затем они сохраняются в файл *params.mat*, заменяются другими значениями и после загрузки отображаются на экране. При выполнении этой программы на экране будут показаны те же значения переменных, которые они принимали в самом начале.

Следует обратить внимание, что функция *load* позволяет загружать из *mat*-файла не все, а только указанные программистом переменные, например:

```
load('params', 'x');
```

загружает только значение переменной *x*.

### **Функции *fwrite* и *fread***

Недостатком рассмотренных функций *save* и *load* является то, что они работают с определенными форматами файлов (обычно *mat*-файлы) и не позволяют загружать или сохранять данные в других форматах. Между тем, бывает необходимость загружать информацию, например, из бинарных файлов, созданных другими программными продуктами для дальнейшей обработки результатов в MATLAB. С этой целью были разработаны функции *fwrite* и *fread*.

В этих функциях используется идентификатор файла — это указатель на файл, с которым предполагается работать. Чтобы получить идентификатор, используется функция *fopen*(имя\_файла, режим\_работы), где параметр режим\_работы может принимать значения, приведенные в следующей таблице:

Режим_работы	Описание
'r'	чтение
'w'	запись (стирает предыдущее содержимое файла)
'a'	добавление (создает файл, если его нет)
'r+'	чтение и запись (не создает файл, если его нет)
'w+'	чтение и запись (очищает прежнее содержимое или создает файл, если его нет)
'a+'	чтение и добавление (создает файл, если его нет)
'b'	дополнительный параметр, означающий работу с бинарными файлами, например, 'wb', 'rb', 'rb+', 'ab' и т.п.

Если функция `fopen` по каким-либо причинам не может корректно открыть файл, то она возвращает значение `-1`. Ниже представлен фрагмент программы записи и считывания данных из бинарного файла:

```
A = [1 2 3 4 5];

fid = fopen('my_file.dat', 'wb');    % открытие файла на запись
if (fid == -1)                       % проверка корректности открытия
    error('File is not opened');
end

fwrite(fid, A, 'double');            % запись матрицы в файл (40 байт)
fclose(fid);                        % закрытие файла

fid = fopen('my_file.dat', 'rb');    % открытие файла на чтение
if (fid == -1)                       % проверка корректности открытия
    error('File is not opened');
end

B = fread(fid, 5, 'double');         % чтение 5 значений double
disp(B);                            % отображение на экране
fclose(fid);                        % закрытие файла
```

В результате работы данной программы в рабочем каталоге будет создан файл *my\_file.dat* размером 40 байт, в котором будут содержаться 5 значений типа `double`, записанных в виде последовательности байт (по 8 байт на каждое значение). Функция `fread` считывает последовательно сохраненные байты и автоматически преобразовывает их к типу `double`, т.е. каждые 8 байт интерпретируются как одно значение типа `double`.

В приведенном примере явно указывалось число элементов (5) для считывания из файла. Однако часто общее количество элементов бывает неизвестным, либо оно меняется в процессе работы программы. В этом случае было бы лучше считывать данные из файла до тех пор, пока не будет достигнут его конец. В MATLAB существует функция для проверки достижения конца файла: `feof(идентификатор_файла)`, которая возвращает 1 при достижении конца файла и 0 в других случаях. Перепишем программу для считывания произвольного числа элементов типа `double` из входного файла.

```
fid = fopen('my_file.dat', 'rb');    % открытие файла на чтение
if (fid == -1)                       % проверка корректности открытия
    error('File is not opened');
end

B = 0;                              % инициализация переменной
cnt = 1;                            % инициализация счетчика
```

```

while (~feof(fid))           % цикл, пока не достигнут конец файла
    [V,N] = fread(fid, 1, 'double'); % считывание одного значения
    if (N > 0)                % если элемент был прочитан успешно, то
        B(cnt) = V;          % формируем вектор-строку из значений V
        cnt = cnt + 1; % увеличиваем счетчик на 1
    end
end
fclose(fid);                 % закрытие файла
disp(B);                     % отображение результата на экран

```

В данной программе динамически формируется вектор-строка по мере считывания элементов из входного файла. MATLAB автоматически увеличивает размерность векторов, если индекс следующего элемента на 1 больше максимального. Однако на такую процедуру тратится много машинного времени и программа начинает работать заметно медленнее, чем если бы размерность вектора B с самого начала была определена равной 5 элементам, например, так

```
B = zeros(1,5);
```

Следует также отметить, что функция `fread` записана с двумя выходными параметрами V и N. Первый параметр содержит значение считанного элемента, а второй — число считанных элементов. В данном случае значение N будет равно 1 каждый раз при корректном считывании информации из файла, и 0 при считывании служебного символа EOF, означающий конец файла.

С помощью функций `fwrite` и `fread` можно сохранять и строковые данные. Например, пусть дана строка

```
str = 'Hello MatLab';
```

которую требуется сохранить в файл. В этом случае функция `fwrite` будет иметь следующую запись:

```
fwrite(fid, str, 'int16');
```

Здесь используется тип `int16`, т.к. при работе с русскими буквами система MATLAB использует двухбайтовое представление каждого символа. Ниже представлена программа записи и чтения строковых данных, используя функции `fwrite` и `fread`:

```

fid = fopen('my_file.dat', 'wb');
if (fid == -1)
    error('File is not opened');
end
str = 'Привет MatLab';           % строка для записи
fwrite(fid, str, 'int16');        % запись в файл
fclose(fid);

fid = fopen('my_file.dat', 'rb');
if (fid == -1)

```

```
        error('File is not opened');  
    end  
    B = fread(fid, 'int16=>char');           % чтение всех данных из файла  
    fclose(fid);                             % закрытие файла  
    disp(B);                                 % отображение строки на экране
```

Результат выполнения программы будет иметь вид:  
Привет MatLab

### Задание к лабораторной работе

1. Прежде чем писать скрипт, составить для данного ряда рекуррентное соотношение, позволяющее вычислить последующий член ряда через его порядковый номер или через предыдущий член ряда.
2. Задать переменную  $x$ , начальные значения для члена ряда и суммы ряда. Организовать цикл расчёта текущего члена ряда и текущей суммы ряда.
3. Цикл продолжать, пока не будет достигнута точность 0,00001. Вывести из цикла в отдельный текстовый файл номер текущего члена ряда, его значение и значение текущей суммы в виде таблицы. После окончания цикла вывести в командное окно полученную сумму.

Вариант	Задание
1	$S = x + \frac{x^5}{5} + \frac{x^9}{9} + \frac{x^{13}}{13} + \dots$
	$S = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$
2	$S = 1 - \frac{3}{2!}x^2 + \frac{9}{4!}x^4 - \frac{19}{6!}x^6 + \dots$
	$S = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots, \quad 0,1 \leq x \leq 0,5$
3	$S = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \frac{x^7}{7 \cdot 7!} + \dots$
	$S = -(1+x)^2 + \frac{(1+x)^4}{2} - \frac{(1+x)^6}{3} + \dots$
4	$S = \sqrt{\frac{2x}{\pi}} \left( \frac{x}{3} - \frac{x^3}{7 \cdot 3!} + \frac{x^5}{11 \cdot 5!} - \frac{x^7}{15 \cdot 7!} + \dots \right)$
	$S = 1 - \frac{1}{1 \cdot 2 \cdot 3 \cdot 4} (2 \cdot 3 \cdot 4 \cdot 5 \cdot x - 3 \cdot 4 \cdot 5 \cdot 6 \cdot x^2 + 4 \cdot 5 \cdot 6 \cdot 7 \cdot x^3 - 5 \cdot 6 \cdot 7 \cdot 8 \cdot x^4 + \dots), \quad  x  < 1$
5	$S = x - \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} - \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots$



	$S = 1 - \frac{x^4}{2^4 \cdot 4^2} + \frac{x^8}{2^4 \cdot 4^2 \cdot 6^4 \cdot 8^2} - \dots$
6	$S = \frac{2}{\sqrt{\pi}} \left[ x - \frac{x^3}{1! \cdot 3} + \frac{x^5}{2! \cdot 5} - \frac{x^7}{3! \cdot 7} + \dots \right]$
	$S = 1 - \frac{5}{2}x + \frac{5 \cdot 7}{2 \cdot 4}x^2 - \frac{5 \cdot 7 \cdot 9}{2 \cdot 4 \cdot 6}x^3 + \dots, \quad  x  < 1$
7	$S = \frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} - \frac{x^9}{63} + \dots$
	$S = 1 + \frac{\cos x}{1!} + \frac{\cos^2 x}{2!} + \frac{\cos^3 x}{3!} + \dots$
8	$S = 0,57722 - \ln x - \frac{x^2}{2 \cdot 2!} + \frac{x^4}{4 \cdot 4!} - \frac{x^6}{6 \cdot 6!} + \dots$
	$S = 0,57722 + \ln(-\ln x) + \frac{\ln^2 x}{2 \cdot 2!} + \frac{\ln^3 x}{3 \cdot 3!} + \dots, \quad 0 < x < 1$
9	$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$
	$S = 1 + \frac{\cos \frac{\pi}{4}}{1!}x + \frac{\left(\cos \frac{\pi}{4}\right)^2}{2!}x^2 + \frac{\left(\cos \frac{\pi}{4}\right)^3}{3!}x^3 + \dots$
10	$S = \frac{\left(\frac{x}{2}\right)^6}{(1!)^2} - \frac{\left(\frac{x}{2}\right)^8}{(3!)^2} + \frac{\left(\frac{x}{2}\right)^{10}}{(5!)^2} - \dots$
	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln^2 3}{2!}x^2 + \frac{\ln^3 3}{3!}x^3 + \dots$
11	$S = \frac{x \cos \frac{\pi}{3}}{1} + \frac{\left(x \cos \frac{\pi}{3}\right)^2}{2} + \frac{\left(x \cos \frac{\pi}{3}\right)^3}{3} + \dots$
	$S = 1 - \frac{\left(\frac{x}{2}\right)^4}{(2!)^2} + \frac{\left(\frac{x}{2}\right)^8}{(4!)^2} - \frac{\left(\frac{x}{2}\right)^{12}}{(6!)^2} + \dots$

12	$S = \sqrt{\frac{2x}{\pi}} \left( 1 - \frac{x^2}{5 \cdot 2!} + \frac{x^4}{9 \cdot 4!} - \frac{x^6}{13 \cdot 6!} + \dots \right)$
	$S = 0,57722 + \ln x + \frac{x}{1 \cdot 1!} + \frac{x^2}{2 \cdot 2!} + \dots$
13	$S = \frac{x^2}{2^2} - \frac{x^6}{2^2 \cdot 4^2 \cdot 6^2} + \frac{x^{10}}{2^2 \cdot 4^2 \cdot 6^2 \cdot 8^2 \cdot 10^2} - \dots$
	$S = 1 - \frac{x^2}{2} + \frac{x^4}{2^2 \cdot 2!} - \frac{x^6}{2^3 \cdot 3!} + \dots$
14	$S = 2 \left( x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right), \quad  x  < 1$
	$S = 1 + \frac{1}{2}x^2 + \frac{1 \cdot 3}{2 \cdot 4}x^4 + \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}x^6 + \dots$
15	$S = 1 + x^2 + \frac{x^4}{2!} + \frac{x^6}{3!} + \dots$
	$S = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots, \quad  x  < 1$
16	$S = x + \frac{x^5}{5} + \frac{x^9}{9} + \frac{x^{13}}{13} + \dots$
	$S = 1 - \frac{x^4}{2^4 \cdot 4^2} + \frac{x^8}{2^4 \cdot 4^2 \cdot 6^4 \cdot 8^2} - \dots$
17	$S = x - \frac{x^3}{3 \cdot 3!} + \frac{x^5}{5 \cdot 5!} - \frac{x^7}{7 \cdot 7!} + \dots$
	$S = 1 - \frac{5}{2}x + \frac{5 \cdot 7}{2 \cdot 4}x^2 - \frac{5 \cdot 7 \cdot 9}{2 \cdot 4 \cdot 6}x^3 + \dots, \quad  x  < 1$
18	$S = x - \frac{1 \cdot x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} - \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots$
	$S = 1 - \frac{x^2}{2} + \frac{x^4}{2^2 \cdot 2!} - \frac{x^6}{2^3 \cdot 3!} + \dots$

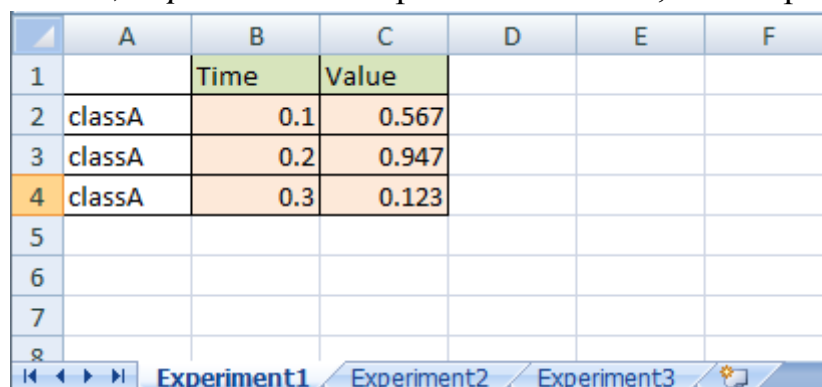
19	$S = \frac{x^3}{3} - \frac{x^5}{15} + \frac{x^7}{35} - \frac{x^9}{63} + \dots$
	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln^2 3}{2!}x^2 + \frac{\ln^3 3}{3!}x^3 + \dots$
20	$S = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$
	$S = 0,57722 + \ln x + \frac{x}{1 \cdot 1!} + \frac{x^2}{2 \cdot 2!} + \dots$

## Файлы Excel

Для считывания данных из Excel в MATLAB есть специальная функция `xlsread`, которая допускает несколько способов вызова. Рассмотрим основные из них.

Указание имени книги Excel в качестве входного аргумента функции `xlsread` приводит к считыванию данных с первого листа книги. Если первый лист не содержит данных, то функция `xlsread` вернет пустой массив. При считывании данных с листа книги Excel игнорируются верхние строки и правые столбцы, содержащие текст. Числовые данные, полученные в ячейках листа в результате вычисления по формулам, считываются функцией `xlsread` как обычные числовые константы, введенные в ячейки.

Если, например, книга *ExpData.xlsx* содержит три листа, которые называются *Experiment1*, *Experiment2*, *Experiment3* и первый лист такой, как на рисунке



	A	B	C	D	E	F
1		Time	Value			
2	classA	0.1	0.567			
3	classA	0.2	0.947			
4	classA	0.3	0.123			
5						
6						
7						
8						

то команда

```
A = xlsread('ExpData.xlsx')
```

приведет к появлению в рабочей среде следующего массива

A =

```
0.1000  0.5670
0.2000  0.9470
0.3000  0.1230
```

Для считывания данных из книги Excel не только с первого, а с произвольного листа, достаточно указать номер листа или его имя в качестве второго входного аргумента. Если в книге *ExpData.xlsx* на втором листе с именем *Experiment2* находятся такие данные

	A	B	C	D	E	F
1		Number	Value	Time		
2	classA	11	1.567	12.345		
3	classA	12	4.947	16.789		
4	classA	13	5.123	18.912		
5						
6						
7						
8						

то для считывания можно использовать команду:

```
A = xlsread('ExpData.xlsx',2)
```

A =

```
11.0000  1.5670  12.3450
12.0000  4.9470  16.7890
13.0000  5.1230  18.9120
```

Если диапазон значений на третьем листе содержит текст и пустые ячейки, то вместо них в возвращаемом функцией `xlsread` массиве будет находиться NaN (не числа — Not a Number). Например, если содержимое третьего листа книги такое

	A	B	C	D	E	F
1		Time			Value	
2	classA	0.1		T=	0.567	
3	classA	0.2		T=	0.947	
4	classA	0.3		T=	0.123	
5						
6						
7						
8						

то результатом работы функции `xlsread` будет массив с четырьмя столбцами и тремя строками:

```
A = xlsread('ExpData.xlsx','Experiment3')
```

A =

```
0.1000  NaN  NaN  0.5670
0.2000  NaN  NaN  0.9470
0.3000  NaN  NaN  0.1230
```

Функция `xlsread` позволяет указать диапазон данных, которые требуется считать. Если диапазон данных указан в качестве ее второго входного аргумента, то происходит считывание данных из этого диапазона на первом листе книги Excel, а если во втором входном аргументе `xlsread` указать имя листа, а диапазон данных — в третьем, то будет производиться считывание данных из заданного

диапазона этого листа. Для считывания данных, расположенных в диапазоне B2:C4, с листа *Experiment2* из предыдущего примера, следует применить:

```
A = xlsread('ExpData.xlsx','Experiment2','B2:C4')
```

```
A =
```

```
11.0000 1.5670
```

```
12.0000 4.9470
```

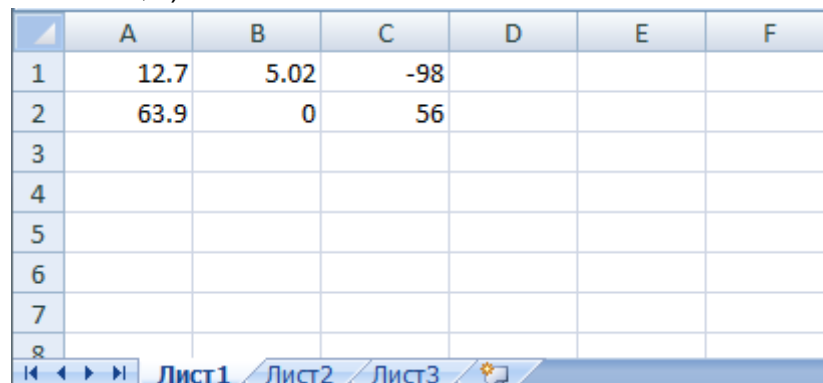
```
13.0000 5.1230
```

Функция `xlsread` предоставляет удобную возможность интерактивного выделения диапазона считываемых данных. Если в качестве ее второго входного аргумента указать -1, то в Excel откроется книга, имя которой задано в первом входном аргументе функции `xlsread`, и появится диалоговое окно. Осталось выбрать лист, на листе выделить диапазон данных и нажать на кнопку ОК в этом окне. Данные из выбранного диапазона вернутся в выходном аргументе функции `xlsread`.

Если требуется записать данные в книгу Excel, то нужно воспользоваться функцией `xlswrite`:

```
A = [12.7 5.02 -98; 63.9 0 56];
```

```
xlswrite('ExpData2.xlsx',A)
```



	A	B	C	D	E	F
1	12.7	5.02	-98			
2	63.9	0	56			
3						
4						
5						
6						
7						
8						

Как и в `xlsread`, во входных аргументах функции `xlswrite` можно указывать номер листа или его имя, диапазон ячеек или начальную ячейку, в которые будут записаны данные. Если нужно записывать данные разного формата, то их нужно собрать в матрицу ячеек.

```
A = {'Time','Temperature'; 12, 98.7; 13, 99.1; 14, 97.8};
```

```
xlswrite('ExpData2.xlsx', A, 2, 'C1')
```

	A	B	C	D	E	F
1			Time	Temperature		
2			12	98.7		
3			13	99.1		
4			14	97.8		
5						
6						
7						
8						

## Файлы изображений

При работе с файлами изображений, представленных в форматах bmp, png, gif, jpeg, tif и т.д., используются функции чтения `imread` и записи `imwrite`.

Ниже приведен пример использования функции `imread` для загрузки растрового изображения

```
[A, map] = imread('1024.bmp','bmp');
```

где `A` — матрица размером `1024x1024xN` точек; `map` — цветовая карта загруженного изображения. Значение `N` показывает число байт, расходуемых на представление точки изображения. Например, если изображение представляется в формате RGB с 24 бит/пиксел, то `N = 3`. Если же загружается изображение с 256 градациями серого (8 бит/пиксел), то `N = 1`.

После обработки изображение `A` можно обратно сохранить в файл, используя следующую запись:

```
imwrite(A, map, 'out_img.bmp', 'bmp');
```

В результате в рабочем каталоге MATLAB будет сохранено изображение в формате bmp с исходной цветовой картой. Однако следует отметить, что если загруженное изображение `A` было преобразовано, например, в формат `double`

```
A = double(A);
```

то непосредственная запись такой матрицы как изображение невозможно. Дело в том, что значения матрицы `A` должны соответствовать целым числам в диапазоне от 0 до 255, т.е. являться байтовыми числами. Этого можно добиться преобразованием типов при записи изображения в файл следующим образом:

```
imwrite(uint8(A), map, 'out_img.bmp', 'bmp');
```

Здесь `uint8` — беззнаковый целый тип в 8 бит.

В качестве переменной `map` можно указывать любые другие цветовые карты (`hot`, `hsv`, `gray`, `pink`, `cool`, `bone` `copper`) отличные от исходной. Например, для записи изображения в 256 градациях серого можно записать

```
imwrite(uint8(A), gray(256), 'out_img.bmp', 'bmp');
```

При этом матрица `A` должна иметь размерность `MxNx1`, т.е. один байт на пиксел.

## **Лабораторная работа 5. Циклы с параметром и работа с матрицами.**

### **Создание матриц**

Самый простой способ формирования векторов и матриц в MATLAB заключается в непосредственном вводе их элементов с клавиатуры. Например, набирая на клавиатуре данные  $X = [1 \ -2 \ 3 \ 8 \ 5 \ 6]$ , получаем одномерный массив (вектор-строку)  $X$  из шести элементов.

Формирование вектора-строки из равноотстоящих значений аргумента выполняется с помощью команды  $x = x0:h:xn$ . По умолчанию шаг  $h$  принимается равным 1. Например, команда  $x = 0:10$  дает целые числа от 0 до 10, а  $x = 0:0.1:10$  задает набор значений аргумента от нуля до 10 с шагом 0.1.

Двумерные массивы задаются в виде матриц, при этом строки разделяются символом «точка с запятой». Элементы одной и той же строки могут разделяться как пробелами, так и запятыми:

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

```
a =
```

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

Для доступа к элементам массива используются круглые скобки:

```
>> a(1,1)
```

```
ans = 1
```

```
>> a(3,3)
```

```
ans = 9
```

```
>> b=[1 2 3 4 5]; b(4)
```

```
ans = 4
```

Для получения строки или столбца матрицы используется символ «двоеточие»:

```
>> a(:,1)
```

```
ans = 1
      4
      7
```

```
>> a(2,:) 
```

```
ans =
      4   5   6
```

```
>> b(1:3)
```

```
ans =
      1   2   3
```

```
>> b(3:end)
```

```
ans =
      3   4   5
```

В MATLAB имеется ряд команд, облегчающих формирование векторов и матриц: `linspace`, `zeros`, `eye`, `logspace`, `ones`, `diag`.

Команда `linspace` используется для получения набора равноотстоящих значений аргумента. Например,  $x = \text{linspace}(x_{\min}, x_{\max})$  создаст массив из 100 значений аргумента между точками  $x_{\min}$  и  $x_{\max}$ . Если требуется иметь другое число точек, например,  $N$  точек на интервале от 0 до 10, следует записать  $x = \text{linspace}(0, 10, N)$ . Аналогичные модификации имеет команда `logspace`, обеспечивающая логарифмическое расположение точек массива.



Для создания некоторых распространенных матриц имеются команды `zeros`, `ones`, `eye` и `diag`. Команды `zeros` и `ones`, создают матрицы, заполненные нулями и единицами соответственно. Так, по командам `A = zeros(3)` и `B = ones(3)` будут сформированы матрицы

```
A= 0 0 0      B = 1 1 1
    0 0 0      1 1 1
    0 0 0      1 1 1
```

а по командам `C = zeros(1,4)` и `D = ones(1,4)` — векторы-строки

```
C = 0 0 0 0      D = 1 1 1 1
```

Команда `eye` предназначена для создания единичной матрицы или ее фрагмента.

```
>> E=eye(3)      >> G=eye(2,7)
E = 1  0  0      G =
    0  1  0      1  0  0  0  0  0  0
    0  0  1      0  1  0  0  0  0  0
```

Команда `diag` имеет два различных назначения. Если ее аргументом является матрица, то результатом будет вектор, содержащий элементы, стоящие на диагонали. Если же аргументом является вектор, то команда `diag` создает матрицу с заданной диагональю и остальными нулевыми элементами. Например, применяя команду `diag` к приведенной выше матрице `B`, получим строку `[1 1 1]`, а повторное применение команды `diag` даст единичную матрицу `E`.

Для создания матрицы из случайных значений используют функции `rand`, `randn` и `randi`. Функция `rand` создает матрицу из равномерно распределенных случайных элементов, находящихся в диапазоне от 0 до 1. Функция `randn` использует нормальное распределение, а `randi` — только целые числа.

Часто возникает необходимость определения общего числа элементов в матрице, т.е. определения её размера. Это можно сделать, воспользовавшись функцией `size` или `length`. Функция `length` возвращает количество элементов в наибольшей размерности матрицы. Например:

```
>>N = length(D)      >> size(G)      >>N = length(G)
N=4                  ans =
                     2    7          N=7
```

### **Объединение и изменение матриц**

Следует отметить, что матрицы можно составлять не только из отдельных чисел или переменных, но и из матриц. Т.е. пара квадратных скобок — это оператор объединения. Например, следующий фрагмент программы показывает, как можно создавать одну матрицу на основе других:

```
>> U = [E randi(9,3); 1 2 3 4 5 6]
```

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

Результатом будет матрица 4x6, получаемая соединением трех подматриц. По вертикали объединять можно только матрицы с одинаковым количеством столбцов.

Для изменения значения элементов матрицы достаточно присвоить им нужное значение:

```
>> E(1:2,2:3) = [5 6; 7 8]
```

```
E =
    1     5     6
    0     7     8
    0     0     1
```

```
E(2,:) = []
```

```
E =
    1     5     6
    0     0     1
```

### **Логическая индексация и функция find**

Логические векторы, созданные из логических операторов и операторов сравнения, могут быть использованы для ссылки на подмассивы. Предположим, что X обыкновенная матрица и L матрица того же размера, но содержащая логические операции. Тогда X(L) задает элементы X, в которых элементы L ненулевые.

Этот вид индексации может быть осуществлен за один шаг указанием логической операции, такой как индексация выражения. Пусть вы имеете следующий набор данных:

```
>> x = [2.1 1.7 1.6 1.5 NaN 1.9 1.8 1.5 5.1 1.8 1.4 2.2 1.6 1.8]
```

NaN — это метка для недостающего наблюдения, как, например, ошибка при ответе на вопрос анкеты. Для того, чтобы убрать данные с логической индексацией, используйте isfinite(x), которая является истиной для всех конечных численных значений и ложью для NaN и Inf.

```
>> x = x(isfinite(x))
```

```
x =
    2.1    1.7    1.6    1.5    1.9    1.8    1.5    5.1    1.8    1.4    2.2    1.6    1.8
```

Сейчас осталась одна наблюдаемая величина, 5.1, заметно отличающаяся от остальных, — это выброс. Последующие действия устраняют выбросы, в данном случае те элементы, для которых среднеквадратичное отклонение более, чем в три раза уклоняется от среднего:

```
>> x = x(abs(x-mean(x)) <= 3*std(x))
```

```
x =
    2.1    1.7    1.6    1.5    1.9    1.8    1.5    1.8    1.4    2.2    1.6    1.8
```

В качестве другого примера обнулим все числа, которые больше 2:

```
>> x(x > 2) = 0
x =
0 1.7 1.6 1.5 1.9 1.8 1.5 1.8 1.4 0 1.6 1.8
```

Функция `find` определяет индексы массива элементов с заданными логическими условиями. Например,

```
>> k = find(x == 1.5)
```

возвращает индексы элементов, которые равны 1.5:

```
k =
4 7
```

Покажем эти элементы, как вектор-строку в порядке, определенном функцией:

```
>> x(k)
ans =
1.5 1.5
```

Если использовать `k` как индекс с левой стороны в операторе присваивания, то матричная структура сохраняется:

```
>> x(k) = 2
x =
0 1.7 1.6 2 1.9 1.8 2 1.8 1.4 0 1.6 1.8
```

Для работы с матрицами удобнее использовать логическую индексацию, т.к. функция `find` преобразует матрицу в вектор индексов и возвращает результат в виде вектора.

### Задание к лабораторной работе

1. Создать mat-файл, в котором записана числовая последовательность из произвольного числа элементов и матрица для второго задания.
2. Ввести из файла и вывести в командное окно исходные данные.
3. Выполнить задачи, применяя оператор цикла с параметром, и вывести в командное окно расчётные данные.
4. Выполнить задачи, не применяя оператор цикла, и вывести в командное окно расчётные данные.

Вариант	Задание
1	Найти произведение отрицательных элементов массива
	Найти наименьшие элементы и номера строк и столбцов, в которых они расположены, целочисленной матрицы $A(6 \times 5)$ . Элементы матрицы $A$ — случайные целые числа из диапазона от 0 до 100
2	Найти сумму положительных элементов массива, стоящих на местах, кратных 5
	Вычислить количество отрицательных элементов каждого столбца для матрицы $B(6 \times 5)$ . Элементы матрицы — случайные целые числа из диапазона от -100 до 100
3	Найти сумму отрицательных элементов массива, расположенных на нечетных местах
	Вычислить сумму и число положительных элементов каждого столбца матрицы $A(8 \times 8)$ , находящихся над главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -100 до 100
4	Найти сумму последних пяти элементов массива, меньших 5
	Вычислить сумму и число положительных элементов каждого столбца для матрицы $B(10 \times 15)$ . Элементы матрицы — случайные целые числа из диапазона от -100 до 100
5	Найти сумму и количество элементов массива, больших единицы

	Вычислить произведение и число отрицательных элементов каждого столбца для матрицы $B(12 \times 15)$ . Элементы матрицы — случайные числа из диапазона от -1 до 1
6	Определить элемент массива, наиболее близкий к заданному числу $x$
	Вычислить суммы и произведения каждой строки матрицы $A(6 \times 5)$ . Элементы матрицы — случайные числа из диапазона от -5 до 5
7	Найти количество нулевых элементов массива среди последних 5 его элементов
	Вычислить сумму и число элементов матрицы $B(6 \times 8)$ , находящихся под главной диагональю и на ней. Элементы матрицы — случайные целые числа из диапазона от -100 до 100
8	Найти произведение положительных элементов массива, расположенных на нечётных местах
	Вычислить сумму и число положительных элементов матрицы $B(8 \times 5)$ , находящихся над главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -50 до 120
9	Найти количество нулевых элементов массива, стоящих на местах, кратных 4
	Вычислить среднее значение и количество отрицательных элементов каждой строки для матрицы $C(6 \times 5)$ . Элементы матрицы — случайные целые числа из диапазона от -15 до 30
10	Найти сумму отрицательных элементов массива, стоящих на чётных местах
	Вычислить среднее и число отрицательных элементов матрицы $C(7 \times 8)$ , находящихся над главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -10 до 10
11	Определить количество элементов массива, кратных 5

	Вычислить среднее и число положительных элементов матрицы $C(8 \times 9)$ , находящихся под главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -20 до 20
12	Найти количество элементов массива, равных 3 и 5
	Вычислить суммы и произведения каждого столбца матрицы $C(7 \times 4)$ . Элементы матрицы — случайные числа из диапазона от -3 до 5
13	Подсчитать сумму и произведение первых 5 положительных элементов массива
	Вычислить среднее и число элементов матрицы $C(6 \times 9)$ , по модулю больших 5. Элементы матрицы — случайные целые числа из диапазона от -16 до 16
14	Найти наибольший отрицательный элемент массива
	Вычислить среднее и число элементов матрицы $C(2 \times 15)$ , по модулю меньших 1. Элементы матрицы — случайные числа из диапазона от -2 до 2
15	Найти сумму элементов массива, больших 5, стоящих на местах кратных трём
	Вычислить сумму и произведение элементов матрицы $C(5 \times 3)$ , находящихся в столбцах с четными номерами. Элементы матрицы — случайные числа из диапазона от 2 до 3
16	Найти произведение положительных элементов массива
	Вычислить сумму и число положительных элементов каждого столбца для матрицы $B(7 \times 5)$ . Элементы матрицы — случайные целые числа из диапазона от -50 до 120
17	Найти сумму отрицательных элементов массива, расположенных на четных местах
	Вычислить суммы и произведения каждой строки матрицы $A(5 \times 6)$ . Элементы матрицы — случайные числа из диапазона от -9 до 9

18	Найти среднее и количество элементов массива, больших единицы
	Вычислить сумму и число отрицательных элементов матрицы $B(6 \times 7)$ , находящихся над главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -10 до 20
19	Найти количество нулевых элементов массива среди последних 7 его элементов
	Вычислить среднее и число положительных элементов матрицы $C(9 \times 9)$ , находящихся над главной диагональю. Элементы матрицы — случайные целые числа из диапазона от -15 до 15
20	Найти количество ненулевых элементов массива, стоящих на местах, кратных 3
	Вычислить суммы и произведения каждого столбца матрицы $C(5 \times 6)$ . Элементы матрицы — случайные числа из диапазона от -5 до 8

## **Массивы ячеек**

Массивы ячеек в MATLAB — это многомерные массивы, элементы которых являются копиями других массивов. Массив ячеек пустых матриц может быть создан с использованием функции `cell`. Но более часто они создаются путем заключения разнообразной группы объектов в круглые скобки. Круглые скобки также используются с индексами для получения доступа к содержанию различных ячеек. Например,

```
>> C = { S sum(S) sum(sum(S)) }
```

дает массив ячеек 1x3. Эти три клетки содержат: магический квадрат, вектор-строку с суммами в столбцах и сумму всех его элементов. Если отобразить C в командной строке, то увидите следующее:

```
C =  
[3x4 double] [1x4 double] [69]
```

Это происходит потому, что первые две ячейки слишком большие для вывода в этом ограниченном пространстве, а третья ячейка содержит только отдельное число, 69, и для него есть необходимая область вывода.

Очень важно запомнить два момента. Первый, для получения содержания одной ячейки, используется индекс в фигурных скобках. Например, `C{1}` возвращает матрицу S, а `C{3}` — 69. Второй, массивы ячеек содержат копии других массивов, а не их указатели. Поэтому, если вы впоследствии измените матрицу S, с C ничего не произойдет.

Трехмерные массивы могут быть использованы для хранения последовательности матриц одинакового размера. А массивы ячеек могут применяться для хранения последовательности матриц различного размера. Например,

```
>> M = cell(3,1);  
>> M{2} = eye(3);  
>> M{1} = zeros(4,5);  
>> M{3} = rand(6,7,2);
```

создает последовательность матриц:

```
M =  
[4x5 double]  
[3x3 double]  
[6x7x2 double]
```

## **Символы и текст**

Для ввода текста в MATLAB, используются одинарные кавычки:

```
s = 'Hello'
```



Результатом не будет являться численная матрица или массив, которые мы рассматривали ранее. Это будет символьный массив 1x5. Внутренне, символы хранятся как числа, но не в формате с плавающей точкой. Оператор

```
a = double(s)
```

преобразует массив символов в числовую матрицу, содержащую представление с плавающей точкой ASCII кода для каждого символа. Результатом будет:

```
a =  
72 101 108 108 111
```

А выражение `s = char(a)` осуществляет обратное превращение.

Соединение квадратными скобками конкатенирует текстовые переменные вместе в большую строку:

```
>> h = [s, ' world']
```

```
h =  
Hello world
```

Оператор `v = [s; 'world']` объединяет строки по вертикали:

```
v =  
Hello  
world
```

Заметьте, что перед символом “w” в переменной `h` необходимо поставить пробел, а оба слова в переменной `v` должны быть равной длины. Результирующие массивы являются снова массивами символов: переменная `h` — 1x11, а переменная `v` — 2x5.

Есть два способа, чтобы управлять группой текста, содержащей строки разной длины: формировать заполненный массив символов или клеточный массив строк. Функция `char` принимает любое число строк, добавляет пробелы в каждую строку, чтобы все они были равной длины, и формирует массив строк с символьной строкой в каждой строке. Например,

```
S = char('A', 'rolling', 'stone', 'gathers', 'momentum.')
```

```
выдает  
S =  
A  
rolling  
stone  
gathers  
momentum.
```

Присутствует достаточное количество пробелов в первых четырех строках, чтобы все строки были равной длины. Другой способ — это сохранить текст в массиве ячеек:

```
>> C = {'A', 'rolling', 'stone', 'gathers', 'momentum.'}
```

```
C =  
'A' 'rolling' 'stone' 'gathers' 'momentum.'
```

Вы можете преобразовать заполненный символьный массив в массив ячеек из строк следующим образом

```
C = cellstr(S)
```

Обратное преобразование

```
S = char(C)
```

## **Структуры**

Структуры — это многомерные массивы MATLAB с элементами, доступ к которым осуществляется через поля. Например,

```
S.name = 'Ed Plum'; S.score = 83; S.grade = 'B+';
```

создает скалярную структуру с тремя полями

```
S =
```

```
    name: 'Ed Plum'
```

```
    score: 83
```

```
    grade: 'B+'
```

Как и всё в MATLAB, структуры являются массивами, поэтому вы можете добавлять в них элементы. В этом случае каждый элемент массива является структурой с несколькими полями. Поля могут добавляться либо по одному

```
S(2).name = 'Toni Miller'; S(2).score = 91; S(2).grade = 'A-' ;
```

либо полностью

```
S(3) = struct( 'name', 'Jerry Garcia', 'score', 70, 'grade', 'C' )
```

Сейчас структура стала достаточно большой, поэтому печатается лишь её сводка

```
S = 1x3 struct array with fields:
```

```
    name
```

```
    score
```

```
    grade
```

Есть несколько способов перетранслировать различные поля в другие массивы MATLAB. Все они базируются на записи списка, разделенного запятыми. Если вы наберете

```
S.score
```

то это будет равносильно следующему

```
S(1).score, S(2).score, S(3).score
```

Это и есть список, разделенный запятыми. Правда, без другой пунктуации он не очень полезен. В этой строке происходит присваивание трех счетов (score) переменной по умолчанию ans и вывод результатов каждого присваивания. Но если вы включаете выражение в квадратные скобки,

```
[S.score]
```

то же самое, что

```
[S(1).score, S(2).score, S(3).score]
```

результатом будет численный вектор-строка, содержащий все счета (score)

```
ans =  
83 91 70
```

Аналогично,  
S.name

просто присваивает имена (names), по одному, переменной ans. Однако, заключение этого выражения в фигурные скобки

```
{S.name}
```

создает массив ячеек 1x3, содержащий три имени (names)

```
ans =  
'Ed Plum' 'Toni Miller' 'Jerry Garcia'
```

И функция

```
char(S.name)
```

с тремя аргументами создает массив символов из поля name:

```
ans =  
Ed Plum  
Toni Miller  
Jerry Garcia
```

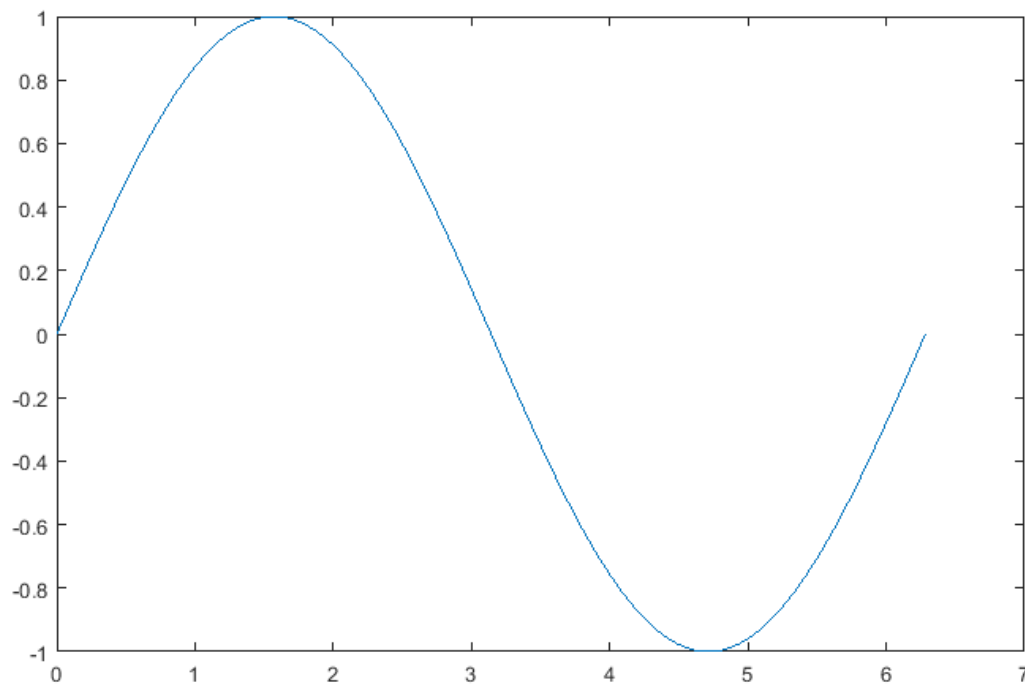
## **Лабораторная работа 6. Графики функций одной переменной.**

### **Создание графика**

Функция `plot` имеет различные формы, связанные с входными параметрами, например `plot(y)` создает кусочно-линейный график зависимости элементов `y` от их индексов. Если вы задаете два вектора в качестве аргументов, `plot(x,y)` создаст график зависимости `y` от `x`.

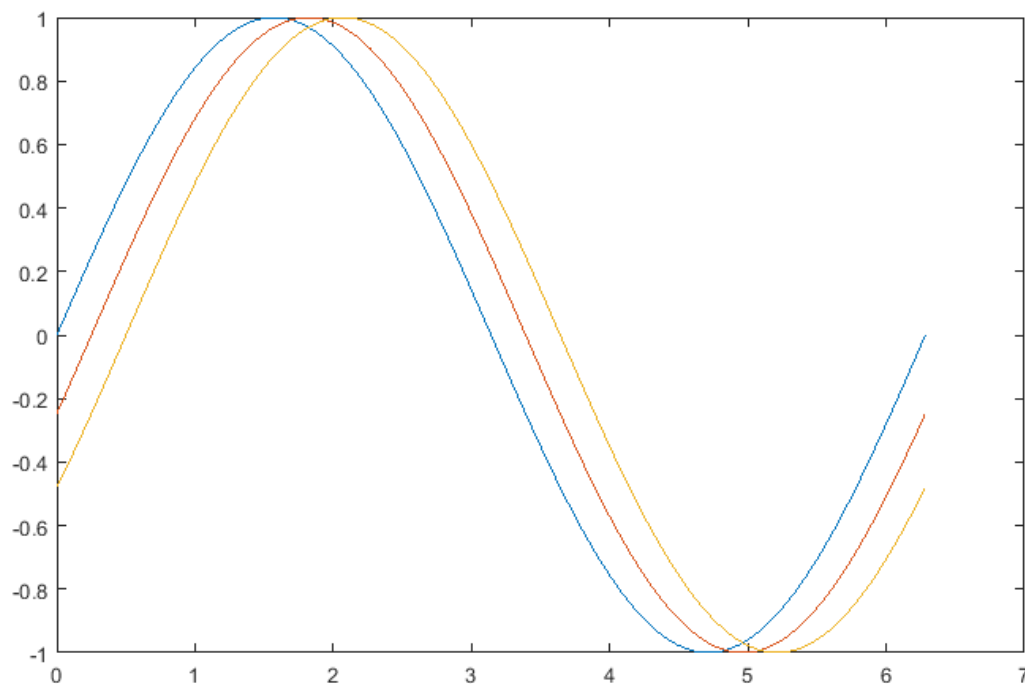
Например, для построения графика значений функции  $\sin$  от нуля до  $2\pi$ , сделаем следующее:

```
t = 0:pi/100:2*pi;  
y = sin(t);  
plot(t,y)
```



Вызов функции `plot` с многочисленными парами `x-y` создает многочисленные графики. MATLAB автоматически присваивает каждому графику свой цвет (исключая случаи, когда это делает пользователь), что позволяет различать заданные наборы данных. Например, следующие три строки отображают график близких функций, и каждой кривой соответствует свой цвет:

```
y2 = sin(t-.25);  
y3 = sin(t-.5);  
plot( t, y, t, y2, t, y3)
```



Возможно изменение цвета, стиля линий и маркеров, таких как знаки плюс или кружки, следующим образом:

`plot(x, y, 'цвет_стиль_маркер')`

цвет\_стиль\_маркер это 1-, 2-, 3- символьная строка (заклученная в одинарные кавычки), составленная из типов цвета, стиля линий и маркеров:

- Символы, относящие к цвету: 'c' , 'm' , 'y' , 'r' , 'g' , 'b', 'w' и 'k'. Они обозначают голубой, малиновый, желтый, красный, зеленый, синий, белый и черный цвета соответственно.

- Символы, относящиеся к типу линий: '-' для сплошной, '--' для разрывной, ':' для пунктирной, '-.' для штрихпунктирной линий и 'none' для её отсутствия.

- Наиболее часто встречающиеся маркеры '+', 'o', '\*', 's', 'v', '^', 'd' и 'x'.

Например, выражение `plot(x,y,'y:')` строит желтый пунктирный график и помещает маркеры '+' в каждую точку данных. Если вы определяете только тип маркера, но не определяете тип стиля линий, то MATLAB выведет только маркеры.

### **Окна изображений**

Функция `plot` автоматически открывает новое окно изображения (далее окно), если до этого его не было на экране. Если же оно существует, то `plot` использует его по умолчанию. Для открытия нового окна и выбора его по умолчанию, наберите

`figure`

Для того, чтобы сделать существующее окно текущим:

`figure(n)`

где  $n$  — это номер в заголовке окна. В этом случае результаты всех последующих команд будут выводиться в это окно.

### **Добавление кривых на существующий график**

Команда `hold` позволяет добавлять кривые на существующий график. Когда вы набираете

`hold on`

MATLAB не стирает существующий график, а добавляет в него новые данные, изменяя оси, если это необходимо. Например, следующий элемент кода вначале создает псевдоцветной график функции `peaks`, а затем накладывает контурные линии той же функции:

```
[x,y,z] = peaks;
```

```
pcolor(x,y,z)
```

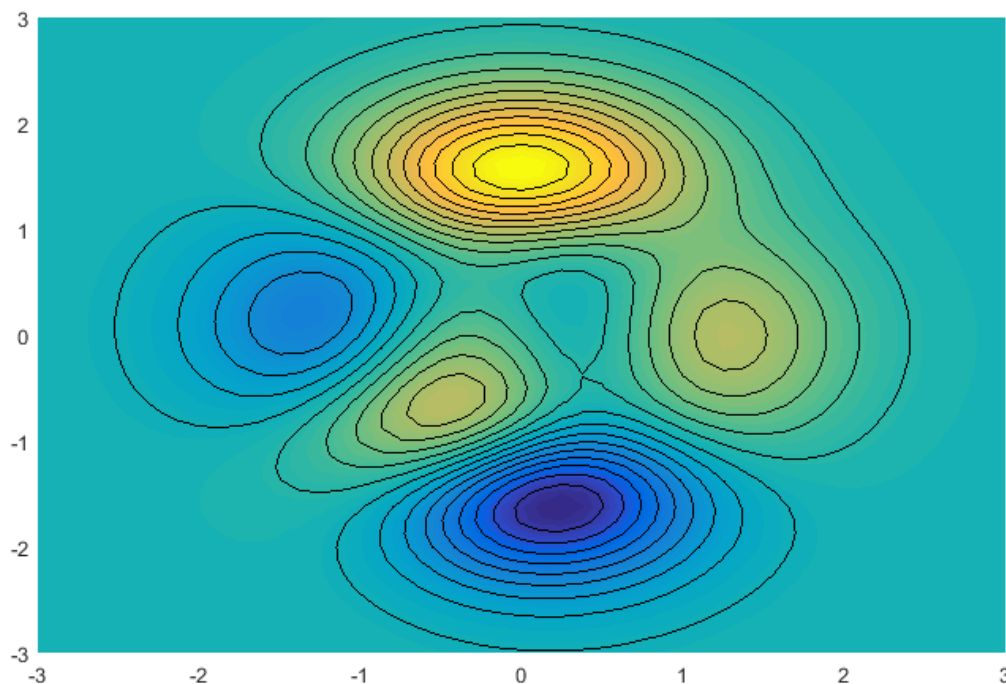
```
hold on
```

```
contour(x,y,z,20,'k')
```

```
shading interp
```

```
hold off
```

Команда `hold on` является причиной того, что график `pcolor` комбинируется с графиком `contour` в одном окне



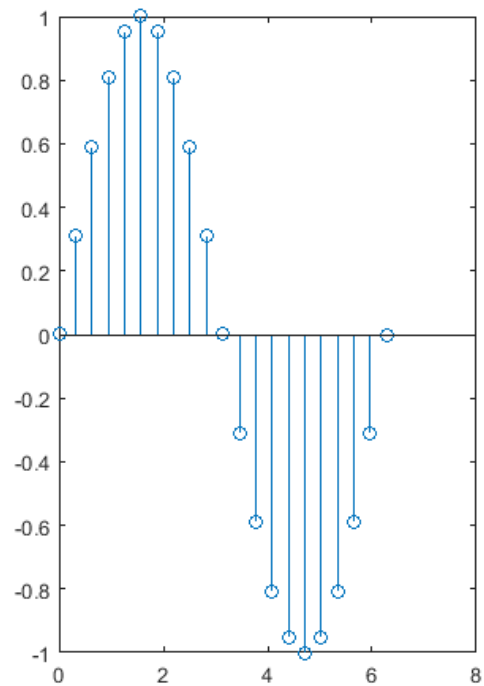
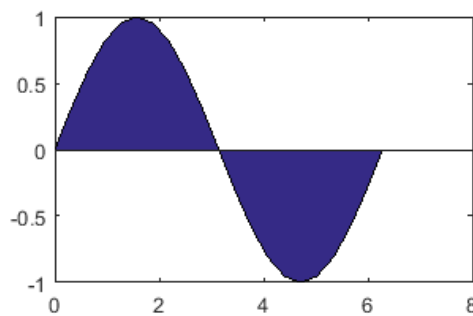
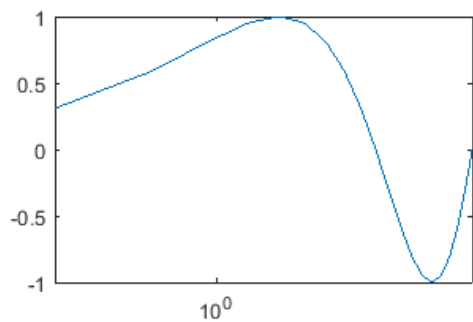
## Подграфики

Функция `subplot` позволяет выводить множество графиков в одном окне или распечатывать их на одном листе бумаги.

`subplot(m,n,p)`

разбивает окно изображений на матрицу  $m$  на  $n$  подграфиков и выбирает  $p$ -ый подграфик текущим. Графики нумеруются вдоль первого в верхней строке, потом во второй и т.д. Например, для того, чтобы представить графические данные в трех разных подобластях окна необходимо выполнить следующее:

```
t = 0:pi/10:2*pi;  
y = sin(t);  
subplot(2,2,1)  
semilog(t,y)  
subplot(2,2,3);  
area(t,y)  
subplot(2,2,[2 4]);  
stem(t,y)
```



## Мнимые и комплексные данные

Если аргумент функции `plot` комплексное число, то мнимая часть игнорируется, за исключением случая, когда комплексный аргумент один. Для этого специального случая происходит построение графика зависимости реальной части аргумента от мнимой. Поэтому `plot(Z)` где  $Z$  комплексный вектор или матрица, эквивалентно

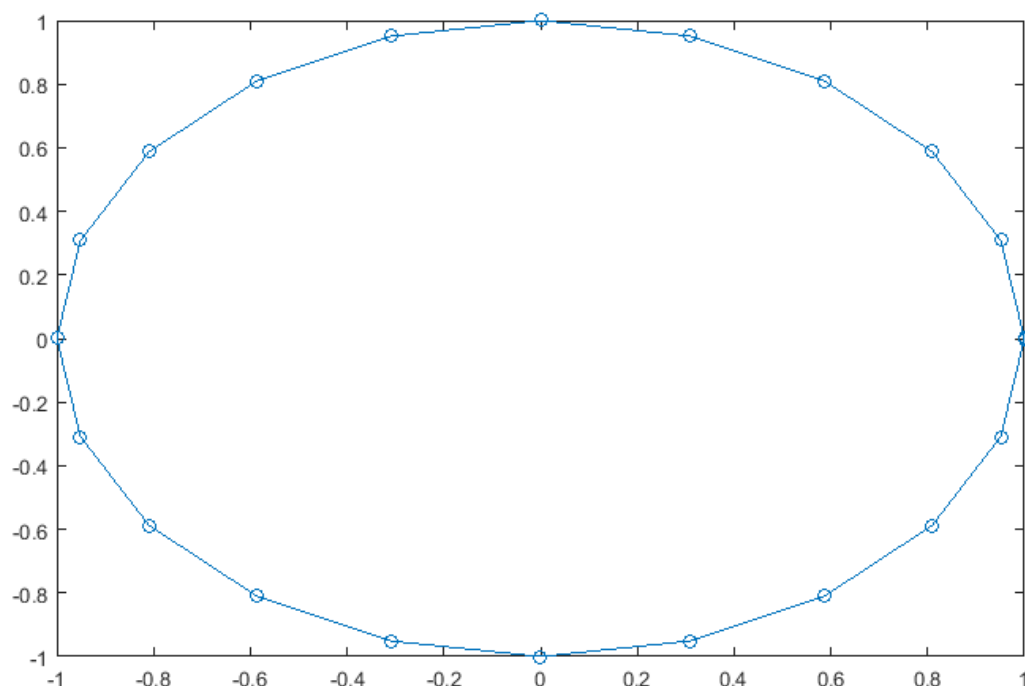
```
plot(real(Z),imag(Z))
```

Например,

```
t = 0:pi/10:2*pi;
```

```
plot(exp(i*t),'-o')
```

отобразит двадцатисторонний многоугольник с маленькими кружками на вершинах:



### Управление осями

Функция `axis` имеет несколько возможностей для настройки масштаба, ориентации и коэффициента сжатия.

Обычно MATLAB находит максимальное и минимальное значение и выбирает соответствующий масштаб и маркеры осей. Функция `axis` заменяет значения по умолчанию предельными значения, вводимыми пользователем.

```
axis( [xmin xmax ymin ymax] )
```

В функции `axis` можно также использовать ключевые слова для управления внешним видом осей. Например,

```
axis square
```

создает x и y оси равной длины, а

```
axis equal
```

создает отдельные отметки приращений для x и y осей одинаковой длины. Так функция

```
plot(exp(i*t))
```



следующая либо за `axis square`, либо за `axis equal` превращает овал в правильный круг.

`axis auto`

возвращает значения по умолчанию и переходит в автоматический режим.

`axis on`

включает обозначения осей и метки промежуточных делений.

`axis off`

выключает обозначения осей и метки промежуточных делений.

`grid off`

выключает сетку координат, а

`grid on`

включает её заново.

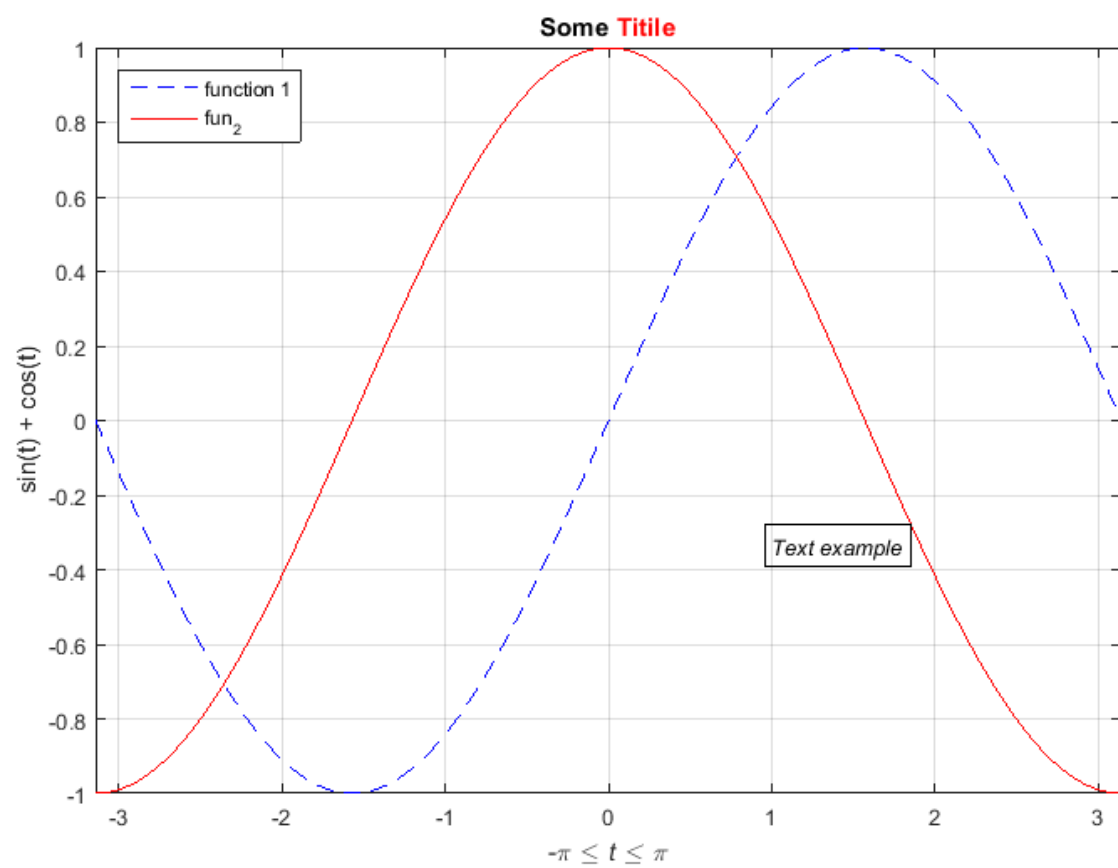
### **Подписи к осям, заголовки и легенда**

Функции `xlabel`, `ylabel`, `zlabel` добавляют подписи к соответствующим осям, функция `title` добавляет заголовок в верхнюю часть окна, а функция `text` вставляет текст в любое место графика. Использование TeX представления позволяет применять греческие буквы, математические символы и различные шрифты.

Функция `legend` добавляет легенду в правый левый угол. В качестве аргументов этой функции используются названия кривых в одинарных кавычках. Используя дополнительные параметры, можно поменять расположение легенды.

Следующий пример демонстрирует эти возможности графики.

```
t = -pi:pi/100:pi;
plot(t,sin(t),'--b');
hold on;
plot(t,cos(t),'r');
hold off;
axis([-pi pi -1 1]);
xlabel('-\pi \leq t \leq \pi');
ylabel('sin(t) + cos(t)');
title('Some \color{red}Title');
text(1, -1/3, '\it{Text example}', 'EdgeColor', 'k');
grid on
legend('function 1', 'fun_2', 'location', 'northwest');
```



### Задание к лабораторной работе

1. Сформировать диапазоны входных значений.
2. Вычислить значения функций на этих диапазонах используя возможности MATLAB по групповой обработке массивов.
3. Подобрать такие параметры вывода, которые позволяют лучше рассмотреть ход кривой графика.
4. Изобразить графики разным цветом, использовать разные маркеры для точек графиков, добавить подписи по осям, сетку и легенду.

Вариант	Задание
1	$y = \begin{cases} \frac{1+x^2}{\sqrt{1+x^4}}, & x \leq 0 \\ 0,2x + \frac{\sin^2 x}{3+x} + 1, & x > 0 \end{cases}$
	$z = \begin{cases} 3 \sin x - \cos^2 x, & x \leq 0 \\ \frac{3\sqrt{1+x^2}}{\ln(x+2)} - 5, & x > 0 \end{cases}$
2	$y = \begin{cases} \frac{3 + \sin^2(2x)}{1 + \cos^2 x}, & x \leq 0 \\ 2x + \frac{\sin^2 x}{3+x}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{3x^2}{1+x^2}, & x \leq 0 \\ \sqrt{1 + \frac{2x}{e^{0,5x} + x^2}}, & x > 0 \end{cases}$
3	$y = \begin{cases} \sqrt{1+x^2}, & x \leq 0 \\ \frac{1+x}{\sqrt[3]{1+e^{-0,2x}} + 1}, & x > 0 \end{cases}$
	$z = \begin{cases} \sqrt{1+2x^2 - \sin^2 x}, & x \leq 0 \\ \frac{2+x}{\sqrt[3]{2+e^{-0,1x}}}, & x > 0 \end{cases}$

4	$y = \begin{cases} \frac{3 + \sin^2 x}{1 + x^2}, & x \leq 0 \\ 2x^2 \cos^2 x, & x > 0 \end{cases}$
	$z = \begin{cases} \sqrt{1 +  x }, & x \leq 0 \\ \frac{1 + 3x}{\sqrt[3]{1 + x} + 2}, & x > 0 \end{cases}$
5	$y = \begin{cases} \frac{\sqrt{1 +  x }}{2 +  x }, & x \leq 0 \\ \frac{1 + x}{2 + \cos^3 x}, & x > 0 \end{cases}$
	$z = \begin{cases} \sqrt[3]{1 + x^2}, & x \leq 0 \\ \sin^2 x + \frac{1 + x}{1 + e^x}, & x > 0 \end{cases}$
6	$y = \begin{cases} \frac{1 +  x }{\sqrt[3]{1 + x + x^2}}, & x \leq -1 \\ \frac{1 + \cos^4 x}{3 + x}, & x > -1 \end{cases}$
	$z = \begin{cases} 2 \ln(1 + x^2), & x \leq -1 \\ (1 + \cos^2 x)^{\frac{3}{5}}, & x > -1 \end{cases}$
7	$y = \begin{cases} \frac{1 + x}{\sqrt[3]{1 + x^2}}, & x \leq 0 \\ 2e^{-2x} - x, & x > 0 \end{cases}$
	$z = \begin{cases} 3x + \sqrt{1 + x^2}, & x \leq 0 \\ 2e^{-2x} \cos x, & x > 0 \end{cases}$
8	$y = \begin{cases} \sqrt{1 + \frac{x^2}{1 + x^2}}, & x \leq 0 \\ 2 \cos x , & x > 0 \end{cases}$
	$z = \begin{cases}  x ^{\frac{1}{3}}, & x \leq 0 \\ \frac{x}{3 + x} - 2x, & x > 0 \end{cases}$

9	$y = \begin{cases} \frac{1+x}{1+x^2}, & x \leq 0 \\ \sqrt{1 + \frac{\cos x}{3+x}}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1+x+x^2}{1+x^2}, & x \leq 0 \\ \sqrt{1 + \frac{2\sin x}{1+x^2}}, & x > 0 \end{cases}$
10	$y = \begin{cases} 1 + \frac{3+x}{1+x^2}, & x \leq 0 \\ \sqrt{1 + (1 - \sin x)^2}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1+2x}{1+x^2}, & x \leq 0 \\ \sin^2 x \sqrt{1+x}, & x > 0 \end{cases}$
11	$y = \begin{cases} \frac{ x }{1+x^2} e^{-2x}, & x \leq 0 \\ \sqrt{1+x^2}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1+x}{1 + \sqrt{ x e^{-x}}}, & x \leq 0 \\ \cos(3x), & x > 0 \end{cases}$
12	$y = \begin{cases} \frac{1+x^2}{1 + \sqrt{ \sin x }}, & x \leq 0 \\ e^{-x} \cos(3x), & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1 + \cos x}{1 + e^{2x}}, & x \leq 0 \\ 1 + \sqrt{1 - (x-1)^2}, & x > 0 \end{cases}$
13	$y = \begin{cases} \frac{e^{-2x}}{1+ x } - 1, & x \leq 0 \\ e^{-3x} \sin(2x), & x > 0 \end{cases}$

	$z = \begin{cases} \frac{2 + \sin x}{1 + \sqrt{1 + x + x^2}}, & x \leq 0 \\ \frac{1}{1 - \sqrt{1 - (x - 1)^2}}, & x > 0 \end{cases}$
14	$y = \begin{cases} \sin x e^{-2x}, & x \leq 0 \\ \frac{x^{\frac{2}{3}}}{1 + x^2}, & x > 0 \end{cases}$
	$z = \begin{cases} \sqrt[4]{1 + e^{3x}}, & x \leq 0 \\ \frac{\cos(5x)}{1 + x^2}, & x > 0 \end{cases}$
15	$y = \begin{cases} \frac{2 + \sin^2 x}{1 + x^2}, & x \leq 0 \\ \frac{4 \cos(3x)}{1 + x^2}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1 + \cos x}{1 + e^{2x}}, & x \leq 0 \\ \frac{1}{1 + \sin(2x)}, & x > 0 \end{cases}$
16	$y = \begin{cases} \frac{1 + x^2}{\sqrt{1 + x^4}}, & x \leq 0 \\ 0,2x + \frac{\sin^2 x}{3 + x} + 1, & x > 0 \end{cases}$
	$z = \begin{cases} \sqrt{1 +  x }, & x \leq 0 \\ \frac{1 + 3x}{\sqrt[3]{1 + x + 2}}, & x > 0 \end{cases}$
17	$y = \begin{cases} \sqrt{1 + x^2}, & x \leq 0 \\ \frac{1 + x}{\sqrt[3]{1 + e^{-0,2x}} + 1}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{2 + \sin x}{1 + \sqrt{1 + x + x^2}}, & x \leq 0 \\ \frac{1}{1 - \sqrt{1 - (x - 1)^2}}, & x > 0 \end{cases}$

18	$y = \begin{cases} \frac{\sqrt{1+ x }}{2+ x }, & x \leq 0 \\ \frac{1+x}{2+\cos^3 x}, & x > 0 \end{cases}$
	$z = \begin{cases}  x ^{\frac{1}{3}}, & x \leq 0 \\ \frac{x}{3+x} - 2x, & x > 0 \end{cases}$
19	$y = \begin{cases} \frac{1+x}{\sqrt[3]{1+x^2}}, & x \leq 0 \\ 2e^{-2x} - x, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1+2x}{1+x^2}, & x \leq 0 \\ \sin^2 x \sqrt{1+x}, & x > 0 \end{cases}$
20	$y = \begin{cases} \frac{1+x}{1+x^2}, & x \leq 0 \\ \sqrt{1+\frac{\cos x}{3+x}}, & x > 0 \end{cases}$
	$z = \begin{cases} \frac{1+\cos x}{1+e^{2x}}, & x \leq 0 \\ 1+\sqrt{1-(x-1)^2}, & x > 0 \end{cases}$

## **Изображения**

Двумерные массивы можно отображать как изображения, где элементы массива определяют их яркость и цвет. Например,

```
load durer  
whos
```

покажет, что файл `durer.mat` в директории `demo` состоит из матрицы размером 648 на 509 (матрицы `X`) и матрицы размером 128 на 3 (матрицы `map`). Элементы матрицы `X` — это целые числа от 1 до 128, которые служат индикаторами в цветном отображении, `map`. Следующие строки

```
image(X)  
colormap(map)  
axis image
```

воспроизводят гравюру Дюрера.

## **Печать и сохранение графики**

Опция `Print` в меню `File` и команда `print` печатают графику MATLAB. Меню `Print` вызывает диалоговое окно, которое позволяет выбирать общие стандартные варианты печати. Команда `print` обеспечивает большую гибкость при выводе выходных данных и позволяет контролировать печать из `m`-файлов. Результат может быть послан прямо на принтер, выбранный по умолчанию, или сохранен в заданном файле. Возможно широкое варьирование формата выходных данных, включая использование `PostScript`.

Например, следующая команда сохранит текущее окно изображения как цветной `png` файл `printed_figure.png`:

```
print('printed_figure','-dpng')
```

Для сохранения графики в файлы также можно использовать функцию `saveas`:

```
saveas(gcf,'Pics/saved_figure.png')
```

где `gcf` — функция, указывающая на активную `figure`.



## Лабораторная работа 7. Многомерные вычисления, 3d-графика.

### Многомерные массивы

Многомерные массивы в MATLAB — это массивы более чем с двумя индексами. Они могут быть созданы вызовом функций `zeros`, `ones`, `rand`, `randn` или `randi` с более чем двумя аргументами. Например,

```
R = randi(5,3,4,2)
```

создает  $3 \times 4 \times 2$  массив с  $3 \times 4 \times 2 = 24$  целыми случайными элементами, не больше 5. Трехмерные массивы можно представить как последовательность матриц  $A(k)$ . В этих случаях, элемент  $(i,j)$   $k$ -ой матрицы обозначается  $A(i,j,k)$ .

```
R(:,:,1) =
```

```
5   3   1   3
1   1   4   1
3   2   2   4
```

```
R(:,:,2) =
```

```
2   4   2   5
4   3   5   3
4   1   1   5
```

Функция `sum(R,d)` вычисляет суммы, изменяя индекс  $d$ . Так `sum(R,1)` — это  $1 \times 4 \times 2$  массив, содержащий 2 вектора-строки:

```
ans(:,:,1) =
```

```
9   6   7   8
```

```
ans(:,:,2) =
```

```
10  8   8  13A
```

А `sum(R,2)` является массивом  $4 \times 1 \times 2$ , содержащим 2 вектора-столбца:

```
ans(:,:,1) =
```

```
12
7
11
```

```
ans(:,:,2) =
```

```
13
15
11
```

И наконец,  $S = \text{sum}(R,3)$  имеет размерность  $3 \times 4 \times 1$ , поэтому он выглядит как матрица  $3 \times 4$ :

```
S =  
    7    7    3    8  
    5    4    9    4  
    7    3    3    9
```

### **Функции mesh и surface**

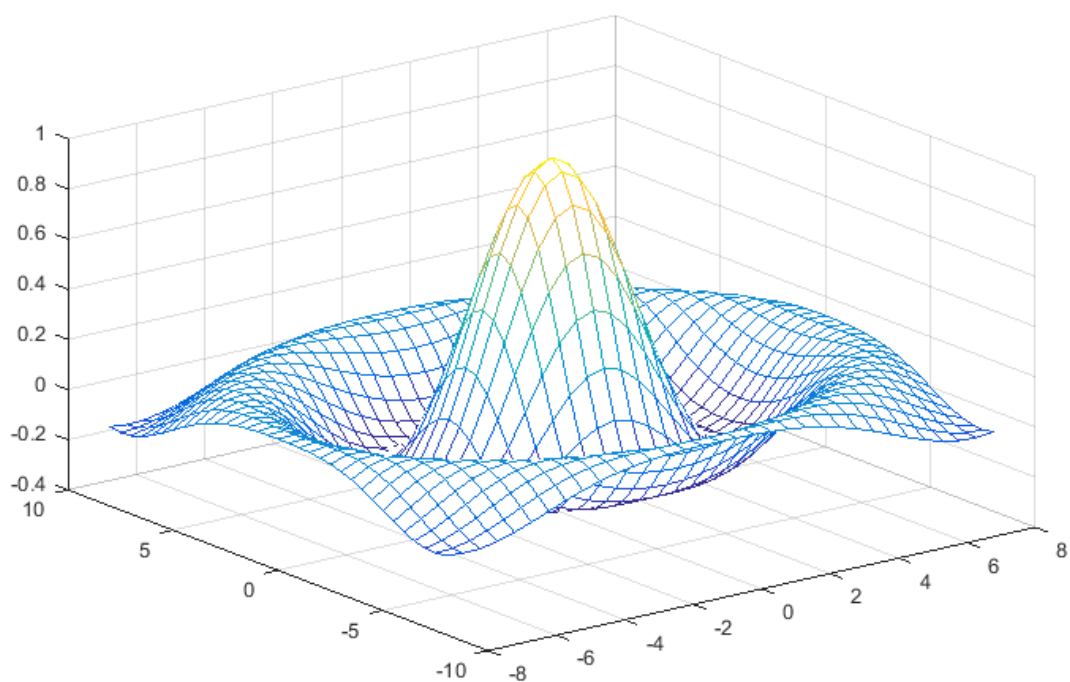
MATLAB определяет поверхность как  $z$  координаты точек над координатной сеткой плоскости  $x$ - $y$ , используя прямые линии для соединения соседних точек. Функции `mesh` и `surface` отображают поверхность в трех измерениях. При этом `mesh` создает каркасную поверхность, где цветные линии соединяют только заданные точки, а функция `surface` вместе с линиями отображает в цвете и саму поверхность.

### **Визуализация функций двух переменных**

Для отображения функции двух переменных,  $z = f(x,y)$ , создаются матрицы  $X$  и  $Y$ , состоящие из повторяющихся строк и столбцов соответственно, перед использованием функции. Затем используют эти матрицы для вычисления и отображения функции. Функция `meshgrid` преобразует область определения, заданную через один вектор или два вектора  $x$  и  $y$ , в матрицы  $X$  и  $Y$  для использования при вычислении функции двух переменных. Строки матрицы  $X$  дублируют вектор  $x$ , а столбцы  $Y$  — вектор  $y$ .

Для построения двумерной функции  $\sin(r)/r$  в области  $x$ - $y$  поступают следующим образом

```
[X, Y] = meshgrid(-8:5:8);  
R = sqrt(X.^2+Y.^2)+eps;  
Z = sin(R)./R;  
mesh(X,Y,Z)
```



В этом примере  $R$  — это расстояние от начала координат, которому соответствует центр матрицы. Добавление  $\epsilon$  позволяет избежать неопределенности  $0/0$  в начале координат.

### Задание к лабораторной работе

1. В скрипте задать исходные данные и вычислить двумерную функцию.
2. Вывести функцию в виде 5 трехмерных графиков разного типа. Каждый график в отдельном окне.
3. Вывести функцию в виде 2 контурных графиков разного типа. Каждый график в отдельном окне.
4. Каждое окно с графиком подписать, в названии указать используемую функцию.

Вариант	Задание
1	$z = \sin x \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
2	$z = \sin \frac{x}{2} \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
3	$z = \sin(2x) \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
4	$z = \sin x \cos \frac{y}{2}, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
5	$z = \sin \frac{x}{2} \cos(2y), \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
6	$z = \sin(2x) \cos(2y), \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
7	$z = \left(1 + \frac{\sin x}{x}\right) \frac{\sin y}{y}, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
8	$z = \frac{\sin x}{x} \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
9	$z = \frac{\sin x}{x}  \cos y , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
10	$z = \frac{\sin x}{x} y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
11	$z = \frac{\sin x}{x}  y , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
12	$z = \frac{\sin x}{x} \sin y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$

13	$z = \frac{\sin x}{x}  \sin y , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
14	$z = \frac{\sin x}{x} (1 - y), \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
15	$z = \frac{\sin x}{x} \left  y + \frac{1}{2} \right , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
16	$z = \sin \frac{x}{2} \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
17	$z = \sin(2x) \cos y, \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
18	$z = \frac{\sin x}{x}  y , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
19	$z = \frac{\sin x}{x}  \cos y , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$
20	$z = \frac{\sin x}{x} \left  y - \frac{1}{2} \right , \quad -2\pi \leq x \leq 2\pi, \quad -2\pi \leq y \leq 2\pi$

## Лабораторная работа 8. Программирование функций.

### Функции

Функции — это m-файлы, которые могут иметь входные и выходные аргументы. Имя m-файла и функции должно быть одним и тем же. Функции работают с переменными в пределах их собственного рабочего пространства, отделенного от рабочего пространства, с которым вы оперируете в командной строке MATLAB.

Хорошим примером является функция rank. m-файл *rank.m* находится в директории toolbox/matlab/matfun. Можно просмотреть его содержание, введя

```
type rank
```

```
function r = rank(A,tol)
%RANK Matrix rank.
% RANK(A) provides an estimate of the number of linearly
% independent rows or columns of a matrix A.
% RANK(A,tol) is the number of singular values of A
% that are larger than tol.
% RANK(A) uses the default tol = max(size(A)) * eps(norm(A)).
%
% Class support for input A:
% float: double, single

% Copyright 1984-2007 The MathWorks, Inc.

s = svd(A);
if nargin==1
    tol = max(size(A)) * eps(max(s));
end
r = sum(s > tol);
```

Первая строка функции m-файла начинается со слова function. Здесь происходит задание имени со списком аргументов. В нашем случае, используется до двух входных аргументов и один выходной.

Следующие несколько строк, до первой пустой или выполняемой строки, являются комментариями, которые предоставляют справочную информацию. Эти строки будут выведены на экран, если вы наберете

```
help rank
```

Первая строка справочного текста — это строка, которую MATLAB отображает при использовании команды lookfor или при запросе help по всей директории.

Остальное содержание файла составляет исполняемый код MATLAB. Переменная *s*, представленная в теле функции, также как и переменные в первой строке, *r*, *A* и *tol*, все являются локальными. Они отделены от других переменных в рабочем пространстве MATLAB.

Этот пример показывает важную особенность функций MATLAB, которая обычно не встречается в других языках программирования, — переменное число аргументов. Функция *rank* может быть использована в нескольких различных формах:

```
rank(A)
r = rank(A)
r = rank(A, 1.e-6)
```

Многие функции MATLAB работают таким образом. Если нет выходного аргумента, то результат сохраняется в переменной *ans*. Если нет второго входного аргумента, то функция вычисляет значение по умолчанию. Внутри тела функции присутствуют две величины *nargin* и *nargout*, которые выдают число входных и выходных аргументов при каждом использовании функции. Функция *rank* использует переменную *nargin*, но не использует *nargout*.

### **Глобальные переменные**

Если вы хотите, чтобы более одной функции использовали отдельную копию переменной, просто объявите её как *global* во всех функциях. Делайте то же самое в командной строке, если вы хотите, чтобы основное рабочее пространство получило доступ к переменной. Определение *global* должно быть до самой переменной, используемой в функции. Хотя это не обязательно, использование больших букв для имени глобальной переменной поможет отличить их от других переменных. Например, создадим m-файл *falling.m*:

```
function h = falling(t)
global GRAVITY
h = 1/2*GRAVITY*t.^2;
```

Затем введем следующие строки

```
global GRAVITY
GRAVITY = 32;
y = falling((0: .1: 5) '');
```

Таким образом, строки определения *GRAVITY* в командной строке делают её доступной внутри функции. Вы можете после изменить *GRAVITY* и получить новое решение, не редактируя какие-либо файлы.

## **Командно-функциональная двойственность**

Примеры команд MATLAB — это `load`, `help`.

Многие команды имеют управляющий параметр, который определяет последующее действие.

```
load August17.dat
```

```
help magic
```

```
type rank
```

Другой метод использования командных параметров — это создание строки аргументов функций.

```
load( 'August17.dat' )
```

```
help( 'magic' )
```

```
type( 'rank' )
```

Это и есть "командно-функциональная двойственность". Любая команда типа

```
command argument
```

также может быть переписана в функциональной форме

```
command( 'argument' )
```

Преимущество функционального подхода проявляется, когда строка аргумента создается из отдельных частей. Следующий пример обрабатывает многочисленные файлы с данными: *August1.dat*, *August2.dat* и т.д. Он использует функцию `int2str`, которая преобразует целые числа в строку символов для создания имени файла.

```
for d = 1:31
```

```
    s = [ 'August' int2str(d) '.dat' ]
```

```
    load(s) % Обработка содержания d-го файла
```

```
end
```

## **Функция eval**

Функция `eval` работает с текстовыми переменными для вычисления и реализации текстовых строк.

```
eval(s)
```

использует интерпретатор MATLAB для вычисления и выполнения выражения, содержащегося в текстовой строке `s`.

Пример из предыдущего раздела может быть также реализован следующим образом (хотя это будет менее эффективно, т.к. используется полный интерпретатор, а не вызов функции).

```
for d = 1:31
```

```
    s = [ 'load August' int2char(n) '.dat' ]
```

```
    eval(s) % Обработка содержания d-го файла
```

```
end
```



## **Векторизация**

Чтобы добиться максимальной скорости вне MATLAB, очень важно векторизовывать алгоритм в m-файлах. Там, где другие языки программирования могут использовать циклы for или do, MATLAB может применять векторные или матричные операции. Простым примером является создание таблицы логарифмов.

```
x = 0
for k = 1:1001
    y(k) = log10(x);
    x = x + .01;
end
```

А векторизованная версия этого кода выглядит следующим образом:

```
x = 0 : 0.1 : 10;
y = log10(x);
```

Для более сложных программ возможности векторизации не так очевидны. Однако, когда важна скорость, вы должны всегда искать способы векторизации вашего алгоритма.

## **Предварительное определение**

Если вы не можете векторизовать часть кода, вы можете заставить ваш цикл for работать быстрее. Для этого нужно предварительно определить векторы или массивы, в которых будут храниться выходные результаты. Например, следующий код использует функция zeros для предварительного определения вектора, создаваемого в цикле for. Это позволяет циклу for работать заметно быстрее.

```
r = zeros(1,32)
for n = 1:32
    r(n) = rank(magic(n));
end
```

Без предварительного определения в предыдущем примере интерпретатор MATLAB увеличивает вектор r по одному элементу каждый раз внутри цикла. Предварительное определение вектора устраняет это действие, и результат получается быстрее.

### Задание к лабораторной работе

1. Задать в скрипте все исходные данные и записать их в файл.
2. Решение задачи сделать внутри функции, используя для вспомогательных вычислений подфункцию.
3. Функция не имеет входных и выходных параметров, а всю необходимую информацию считывает и записывает в файлы, названия которых — глобальные переменные.
4. Подфункция принимает все необходимые данные через входные параметры от функции, и выводит расчётные значения через выходные параметры обратно в функцию.
5. Скрипт выводит расчетные значения из файла в командное окно.

Вариант	Задание
1	Дана матрица. Сумму элементов каждой строки записать в один одномерный массив, а произведение элементов каждого столбца — в другой.
	Из элементов одномерного массива, попавших в интервал $[a,b]$ , найти минимальный элемент.
2	Из одномерного массива исключить максимальный элемент. Вывести номер исключённого элемента и преобразованный массив.
	Из элементов одномерного массива, стоящих на чётных местах и расположенных правее минимального элемента, сформировать новый массив.
3	В одномерном массиве поменять местами максимальный элемент с последним отрицательным элементом.
	В одномерном массиве найти минимальный элемент среди элементов, расположенных между первым и последним нулевыми элементами.
4	Заменить все элементы матрицы, не попавшие в интервал $[a,b]$ , на среднее из всех положительных её значений.
	В одномерном массиве найти максимальный из элементов, стоящих на чётных местах. Умножить на него все элементы

	данного массива, стоящие на нечётных местах и расположенные правее найденного максимального.
5	Дан одномерный массив. Найти сумму элементов до максимального элемента и сумму элементов, расположенных правее него.
	Из данного одномерного массива сформировать новый массив, в который записать числа, находящиеся в массиве между его максимальным и минимальным (или минимальным и максимальным) элементами.
6	Пять последних положительных элементов одномерного массива умножить на номер максимального элемента из этого массива.
	Из одномерного массива сформировать два массива: в один записать элементы, расположенные до минимального элемента, а в другой — расположенные правее минимального.
7	Дана матрица. Сумму максимального и минимального элементов каждой строки записать в одномерный массив.
	В одномерном массиве найти номер третьего нулевого элемента, расположенного за максимальным элементом.
8	В одномерном массиве найти максимальный элемент среди элементов, расположенных после третьего нулевого.
	Найти сумму положительных элементов одномерного массива, расположенных до первого нулевого, заменить этой суммой минимальный элемент.
9	Найти минимальный элемент среди элементов одномерного массива, расположенных после второго положительного элемента.
	Среди элементов одномерного массива, расположенных до первого отрицательного, найти минимальный элемент. Из положительных элементов этого массива, расположенных правее минимального, сформировать массив.

10	В одномерном массиве заменить нулём все отрицательные элементы, предшествующие его максимальному элементу.
	Из элементов одномерного массива, расположенных между первым нулевым и максимальным, сформировать новый массив.
11	Заменить все отрицательные элементы матрицы на её среднеарифметическое значение.
	Дан одномерный массив. Умножить каждый положительный элемент на квадрат его наименьшей компоненты и каждый отрицательный элемент на квадрат его наибольшей компоненты.
12	Найти номер первого нулевого элемента одномерного массива и произведение элементов, расположенных до него. Среди элементов, расположенных правее этого элемента, найти максимальный элемент.
	В одномерном массиве найти количество нулей, находящихся между максимальным и минимальным элементами.
13	Три отрицательных элемента одномерного массива, расположенных правее максимального, умножить на номер максимального элемента.
	В данном одномерном массиве поменять местами минимальный и максимальный элементы.
14	Из одномерного массива получить массив, состоящий из элементов этого массива, расположенных правее его максимального элемента.
	Найти сумму элементов одномерного массива, стоящих правее первого положительного элемента, максимальный элемент и его номер среди чисел, предшествующих первому положительному.
15	Дана квадратная матрица и одномерный массив, число элементов которого равно числу строк матрицы. Изменить матрицу так, чтобы к элементам её столбцов добавить

	соответствующие элементы из массива. Затем из элементов строк полученной матрицы вычесть соответствующие элементы массива.
	В одномерном массиве поменять местами минимальный элемент с первым положительным элементом.
16	Дана матрица. Сумму элементов каждого столбца записать в один одномерный массив, а произведение элементов каждой строки — в другой.
	В одномерном массиве найти максимальный из элементов, стоящих на нечётных местах. Умножить на него все элементы данного массива, стоящие на чётных местах и расположенные правее найденного максимального.
17	В одномерном массиве поменять местами минимальный элемент с первым нулевым элементом.
	Из одномерного массива сформировать два массива: в один записать элементы, расположенные до максимального элемента, а в другой — расположенные левее максимального.
18	Заменить все положительные элементы матрицы на её среднеарифметическое значение.
	Найти сумму отрицательных элементов одномерного массива, расположенных до первого нулевого, заменить этой суммой максимальный элемент.
19	Дана матрица. Сумму максимального и минимального элементов каждой строки записать в одномерный массив.
	В одномерном массиве поменять местами минимальный элемент с первым отрицательным элементом.
20	Найти максимальный среди элементов одномерного массива, расположенных после второго положительного элемента.
	В одномерном массиве найти количество нулей, находящихся между максимальным и минимальным элементами.

## **Лабораторная работа 9. Решение уравнений в MATLAB.**

### **Решение нелинейных уравнений**

Одна из распространенных алгебраических задач — поиск корней уравнения  $f(x) = 0$ . Для численного отыскания корней проще всего построить график функции  $y = f(x)$  и найти точки его пересечения с осью абсцисс. Аналогично можно поступить и в случае системы двух уравнений с двумя неизвестными  $f(x,y) = 0$ ,  $g(x,y) = 0$ , построив на плоскости  $(x,y)$  графики этих функций и найдя точки их пересечения друг с другом. Сложнее обстоит дело с поиском комплексных корней, здесь требуется привлечение специальных методов.

Для поиска корней нелинейных уравнений с одной переменной, например, включающих логарифмические, тригонометрические, экспоненциальные зависимости, применяют функцию `fzero`. Ее входными аргументами служат имя функции, вычисляющей левую часть уравнения  $f(x) = 0$ , и начальное приближение  $x_0$ .

В качестве примера возьмем полином  $f(x) = x^2 + 3x + 2$  с корнями  $-1$  и  $-2$ . В MATLAB их можно найти двумя путями: посредством функции `roots`

```
p = [1 3 2];  
roots(p)  
ans = -2  
      -1
```

и с помощью функции `fzero`. В последнем случае нужно сформировать вспомогательную функцию `ff` в виде отдельного `m`-файла

```
function y = ff(x)  
y = x^2+3*x+2;  
  
fzero(@ff, 0)  
ans = -1
```

или же создать временную функцию (на период данного сеанса MATLAB) при помощи команды `@`.

```
f = @(x) x^2 + 3*x + 2  
fzero(f, -3)  
ans = -2  
  
fzero(f, 0)  
ans = -1
```

Функция `fzero`, использующая численные методы, возвращает один из корней полинома в зависимости от начального приближения.

## Нахождение экстремумов

Для численного решения задач поиска экстремумов функций одного или нескольких аргументов применяют методы половинного деления и золотого сечения (одномерный поиск), методы градиента и наискорейшего спуска, методы целочисленного, линейного и нелинейного программирования. В пакете MATLAB эти методы реализованы в функциях `fminsearch`, `fminunc`, `fmincon`, `fminbnd`. Для задач линейного программирования предназначена функция `linprog`.

Минимум одномерной функции отыскивают с помощью команды `fminsearch`. Для поиска максимума функции  $f(x)$  достаточно найти минимум функции  $-f(x)$ , поэтому специальной функции для поиска максимумов функций в MATLAB не существует.

Найдем точку минимума полинома  $f(x) = x^2 + 3x + 2$  из предыдущего примера:

```
fminsearch(f, 0)
ans = -1.5
```

Убедимся в правильности результата, приравнивая производную функции  $f$  нулю. Производную берем с помощью функций `polyder` и `polyval`, ее корень находим командой `fzero`:

```
polyder(p)
ans = 2 3
df = @(x) polyval(pp,x);
fzero(df,0)
ans = -1.5
```

Расчет значения функции  $f(x)$  в точке минимума:

```
f(ans)
ans = -0.25
```

В качестве ещё одного примера найдем минимум нелинейной функции `humps` из директории `/toolbox/matlab/demos/`. Ниже приведена упрощенная версия функции `humps`:

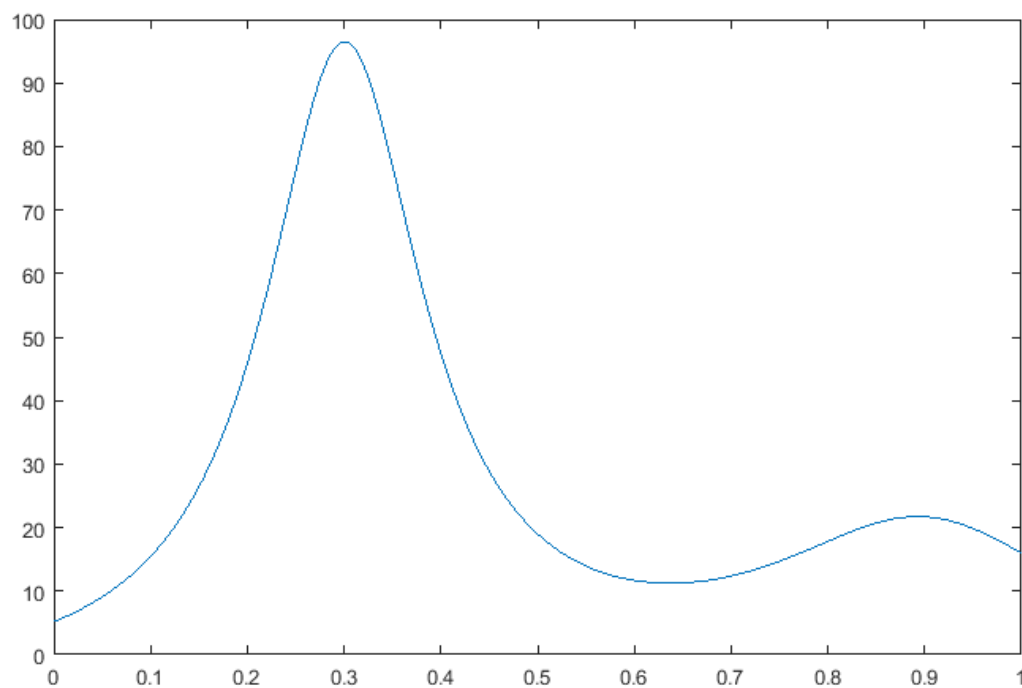
```
function y = humps(x)
y = 1 ./ ((x-.3).^2 + .01) + 1 ./ ((x-.9).^2 + .04) - 6;
```

Вычислим эту функцию в нескольких точках интервала  $[0, 1]$

```
x = 0 : 0.002 : 1;
y = humps(x);
```

Затем построим график

```
plot(x,y)
```



На графике видно, что эта функция имеет локальный минимум около  $x=0.6$ .  
Используем функцию `fminsearch`:

```
p = fminsearch( 'humps', 0.5)  
p =  
0.6370
```

Чтобы найти значение функции в минимуме, запишем следующий код:

```
humps(p)  
ans =  
11.2528
```



### Задание к лабораторной работе

1. Создать функцию  $f_1(x)$ . Вывести  $y(x) = f_1(x)$  в виде графика. По нему определить корни уравнения. Если корни не просматриваются, то изменить пределы изменения аргумента и повторить операции.
2. Для каждого корня из отрезка  $[-10; 10]$  найти точное значение, используя функцию `fzero`. Для этого, в качестве второго аргумента функции `fzero` задать приблизительное значение корня.
3. Сформировать строку с результатами и вывести ее в заголовок окна графика.
4. Создать функции для  $f_2(x)$  и  $f_3(x) = f_1(x) - f_2(x)$ . Вывести  $f_1(x)$  и  $f_2(x)$  в виде графиков. По ним определить приблизительные корни системы уравнений. Если корни на графике не просматриваются, то изменить пределы изменения аргумента и повторить операции.
5. Для каждого корня в отрезке  $[-10; 10]$  найти точное значение, сформировать строку с результатами и вывести ее в заголовок окна графика.

Вариант	Задание
1	$f_1(x) = -x^2 + 4x + 11$
	$f_2(x) = 0.2e^x - 20$
2	$f_1(x) = 2x^2 - 2x - 15$
	$f_2(x) = 40 \cos x $
3	$f_1(x) = x^2 - 4x - 1$
	$f_2(x) = 10 \ln(x + 25)$
4	$f_1(x) = 9x^2 - 8x - 70$
	$f_2(x) = 100 \sin x $
5	$f_1(x) = -4x^2 + 4x + 50$
	$f_2(x) = 70 \cos x - 30$
6	$f_1(x) = 0.1x^3 - 5x^2 + 4x + 40$
	$f_2(x) = 60e^{ 0.1x } - 100$

7	$f_1(x) = 0.2x^3 - 3x^2 + 2x + 30$
	$f_2(x) = 20 \sin(2x) + 8$
8	$f_1(x) = 0.3x^3 - 6x^2 + x + 50$
	$f_2(x) = xe^{ x } + 20$
9	$f_1(x) = 0.4x^3 - 9x^2 + x + 70$
	$f_2(x) = e^{ x } + \cos(3x)$
10	$f_1(x) = 0.5x^3 - 7x^2 + 5x + 60$
	$f_2(x) = -60 \cos x  + 15$
11	$f_1(x) = -0.1x^3 - 4x^2 + 9x + 60$
	$f_2(x) = 15 \log(x + 15)$
12	$f_1(x) = -0.2x^3 - 6x^2 - 7x + 55$
	$f_2(x) = 10 \ln(15 - x)$
13	$f_1(x) = -0.3x^3 - 9x^2 - 8x + 75$
	$f_2(x) = -100 \cos x $
14	$f_1(x) = -0.4x^3 + 7x^2 + 8x - 75$
	$f_2(x) = 100 \sin(0.5x)$
15	$f_1(x) = -0.8x^3 + 8x^2 + 4x - 1$
	$f_2(x) = 10 \cos(0.5x) - 1$
16	$f_1(x) = -x^2 + 4x + 7$
	$f_2(x) = 0.2e^x - 20$
17	$f_1(x) = x^2 + 4x - 3$
	$f_2(x) = 10 \ln(x + 17)$
18	$f_1(x) = 0.2x^3 - 6x^2 + x + 30$

	$f_2(x) = e^{ x } + \sin(2x)$
19	$f_1(x) = -0.1x^3 - 5x^2 + 9x + 32$
	$f_2(x) = 15 \log(2 + \cos(x))$
20	$f_1(x) = -0.2x^3 - 3x^2 + 2x + 30$
	$f_2(x) = 20 \sin(2x) - 31$

## **Лабораторная работа 10. Графические интерфейсы, формы, функции обработки событий.**

### **Графические объекты**

Графические объекты — это базисные элементы системы управляемой графики в MATLAB. Они сформированы в дерево структурной иерархии. Этим отражается связь графических объектов. Например, объекты Line (линия) нуждаются в объектах Axes (оси) как в системе отсчета. В свою очередь объекты Axes существуют только с объектами Figure.

Основные виды объектов управляемой графики:

- Объекты Root являются вершиной иерархии. Они соответствуют экрану компьютера. MATLAB автоматически их создает в начале сеанса работы.
- Объекты Figure — это окна на экране, кроме командного окна.
- Объекты Uicontrol — это пользовательское управление интерфейсом. Когда пользователь активирует объект, вызывается соответствующая функция. Они включают в себя такие элементы как pushbutton, radio button и slider.
- Объекты Axes определяют область в окне Figure и ориентацию дочерних объектов в этой области.
- Объекты Uimenu представляют собой меню пользовательского интерфейса, которое расположено в верхней части окна Figure.
- Объекты Image — это двумерные объекты, которые выводит MATLAB, используя элементы прямоугольного массива как индексы в палитре.
- Объекты Line являются основными графическими базисными элементами для большинства двумерных графиков.
- Объекты Surface — это трехмерное представление данных матрицы, созданное путем графического отображения данных как высот над плоскостью x-y.
- Объекты Text — это строки символов.
- Объекты Light определяют источник света, действующий на все объекты в пределах Axes.

### **Управление объектами**

Каждый отдельный графический объект имеет свой уникальный идентификатор, называемый handle (манипулятор), который MATLAB присваивает объекту при создании. Некоторые графики, например с несколькими кривыми, состоят из многих объектов, каждый из которых имеет свой собственный идентификатор (handle). Чем пытаться прочесть их с экрана

и повторно вводить, вы увидите, что всегда лучше хранить значение в переменной и использовать его по необходимости.

Идентификатор объекта `root` всегда нуль. Идентификатор `figure` — это целое число, которое по умолчанию отображается в заголовке окна. Идентификаторами других объектов являются числа с плавающей точкой, которые содержат информацию, используемую MATLAB. Например, если `A` — это магический квадрат Дюрера, то

```
h = plot(A)
```

создаст линейный график с четырьмя линиями, для каждого столбца `A`, а также возвратит вектор-идентификатор четырёх линий:

```
h =  
4x1 Line array
```

Идентификаторы являются несущественными и могут изменяться от одной системы к другой. Важным является то, что `h(1)` — это идентификатор первой линии на графике, `h(2)` — второй и т.д.

MATLAB имеет несколько функций для получения доступа к часто используемым объектам:

`gcf` — идентификатор активной, последней использованной `figure`,

`gca` — идентификатор активных, последних использованных `axes`.

Вы можете использовать эти функции в качестве входных аргументов других функций, оперирующих с идентификаторами `figure` и `axes`. Получить идентификаторы других объектов можно во время их создания. Все функции MATLAB, которые создают объекты, возвращают идентификатор (или вектор идентификаторов) созданного объекта.

Для удаления объекта служит функция `delete`, использующая идентификатор объекта в качестве аргумента. Например, удалим текущие оси (`axes`), а вместе с ними и все дочерние объекты.

```
delete(gca)
```

### **Функции создания объектов**

Вызов функции с именем какого-либо объекта создает этот объект. Например, функция `text` создает объект `text`, функция `figure` создает объект `figure` и т.д. Графические функции MATLAB высокого уровня (такие как `plot` и `surface`) вызывают необходимые функции низкого уровня для отображения соответствующих графиков.

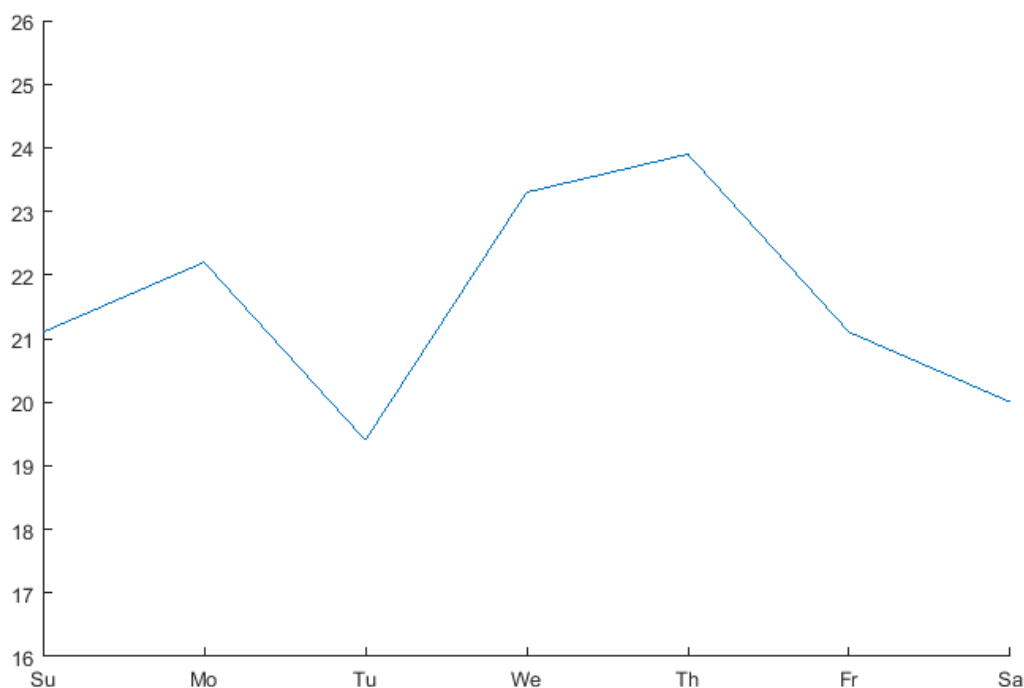
Функции низкого уровня просто создают один из графических объектов определенных MATLAB, за исключением корневого объекта (`root`), который создает только MATLAB. Например,

```
line([1 3 6], [8 -2 0], 'Color', 'red')
```

## Свойства объекта

Все объекты обладают свойствами, которые определяют, как они выводятся на экран. MATLAB предоставляет два механизма для задания свойств. Свойства объекта могут быть установлены функцией создания объекта или могут быть изменены функцией `set`, когда объект уже существует. Например, следующий элемент кода создает три объекта и заменяет некоторые из их свойств по умолчанию.

```
days = [ 'Su' ; 'Mo' ; 'Tu' ; 'We' ; 'Th' ; 'Fr' ; 'Sa' ];  
temp = [ 21.1 22.2 19.4 23.3 23.9 21.1 20.0 ];  
f = figure;  
a = axes( 'YLim' , [16 26] , 'Xtick' , 1:7 , 'XtickLabel' , days );  
h = line(1:7, temp);
```



`days` — это массив символов, содержащий сокращения дней недели, а `temp` — это численный массив с типичной температурой. Окно изображений создается после вызова `figure` без аргументов, т.е. со значениями по умолчанию. Оси существуют внутри `figure` и имеют заданный диапазон по `y` и заданные метки для приращений по `x`. Линии существуют внутри осей и имеют заданные значения для данных по `x` и `y`. Три идентификатора объектов `f`, `a` и `h` сохранены для дальнейшего использования.

## **Функции set и get**

Свойства объекта задаются обращением к нему после его создания. Для этого используйте идентификатор, возвращаемый создаваемой функцией.

Функция `set` позволяет устанавливать свойства объектов, указанием идентификатора объекта и совокупности пар название свойства / значение. В качестве примера изменим цвет и ширину линии:

```
set( h , 'Color' , [0 0.8 0.8] , 'LineWidth' , 3)
```

Для того, чтобы увидеть список всех доступных свойств заданного объекта, вызовите функцию `set` с идентификатором объекта:

```
set(h)
```

```
AlignVertexCenters: {'on' 'off'}
BusyAction: {'queue' 'cancel'}
ButtonDownFcn: {}
Children: {}
Clipping: {'on' 'off'}
Color: {1x0 cell}
CreateFcn: {}
DeleteFcn: {}
DisplayName: {}
HandleVisibility: {'on' 'callback' 'off'}
HitTest: {'on' 'off'}
Interruptible: {'on' 'off'}
LineStyle: {'-' '--' ':' '-.' 'none'}
LineWidth: {}
Marker: {'+' 'o' '*' '.' 'x' 'square' 'diamond' 'v' '^' '>' '<' 'pentagram' 'hexagram'
'none'}
MarkerEdgeColor: {'none' 'flat' 'auto'}
MarkerFaceColor: {'none' 'flat' 'auto'}
MarkerSize: {}
Parent: {}
PickableParts: {'visible' 'none' 'all'}
Selected: {'on' 'off'}
SelectionHighlight: {'on' 'off'}
Tag: {}
UIContextMenu: {}
UserData: {}
Visible: {'on' 'off'}
XData: {}
YData: {}
ZData: {}
```

Чтобы вывести список всех текущих установленных свойств заданного объекта, вызовите функцию `get` с идентификатором объекта:

```
get(h)
```

```
AlignVertexCenters: 'off'  
Annotation: [1x1 matlab.graphics.eventdata.Annotation]  
BeingDeleted: 'off'  
BusyAction: 'queue'  
ButtonDownFcn: ''  
Children: []  
Clipping: 'on'  
Color: [0 0.4470 0.7410]  
CreateFcn: ''  
DeleteFcn: ''  
DisplayName: ''  
HandleVisibility: 'on'  
HitTest: 'on'  
Interruptible: 'on'  
LineStyle: '-'  
LineWidth: 0.5000  
Marker: 'none'  
MarkerEdgeColor: 'auto'  
MarkerFaceColor: 'none'  
MarkerSize: 6  
Parent: [1x1 Axes]  
PickableParts: 'visible'  
Selected: 'off'  
SelectionHighlight: 'on'  
Tag: ''  
Type: 'line'  
UIContextMenu: []  
UserData: []  
Visible: 'on'  
XData: [1 2 3 4 5 6 7]  
YData: [21.1000 22.2000 19.4000 23.3000 23.9000 21.1000 20]  
ZData: [1x0 double]
```

Для запроса значения отдельного свойства используйте `get` с именем свойства:

```
get(h, 'Color')  
ans =  
0 0.8000 0.8000
```



Объекты axes имеют много детальных свойств для всего графика. Например, заголовок — title.

```
t = get(a , 'title' );
```

```
set(t , 'String' , 'Temperature' , 'FontAngle' , 'oblique')
```

Функция title обеспечивает другой интерфейс к этим же свойствам.

## **Графический Пользовательский Интерфейс (GUI)**

Ниже приведен пример, иллюстрирующий использование управляемой графики (Handle Graphics) для создания пользовательского интерфейса.

```
b = uicontrol( 'Style' , 'pushbutton' , 'Units' , 'normalized' , ...
```

```
'Position' , [0.5 0.5 0.2 0.1] , 'String' , 'click here' );
```

создает pushbutton в центре окна изображения (figure) и возвращает идентификатор нового объекта. Однако пока нажатие на эту кнопку ни к чему не приводит.

```
s = ' set(b , 'Position' , [0.8*rand 0.9*rand 0.2 0.1]) ';
```

создает строку, содержащую команду, которая меняет положение кнопки.

Повторное использование

```
eval(s)
```

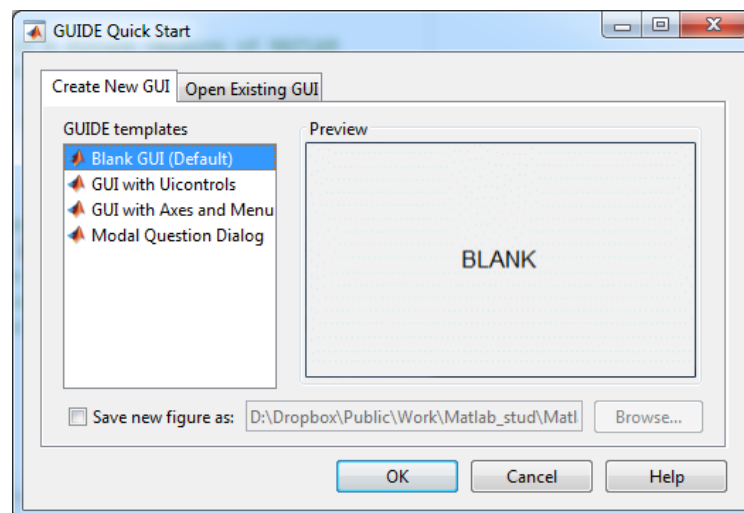
будет передвигать кнопку в случайные места. Окончательно,

```
set(b , 'Callback' , s)
```

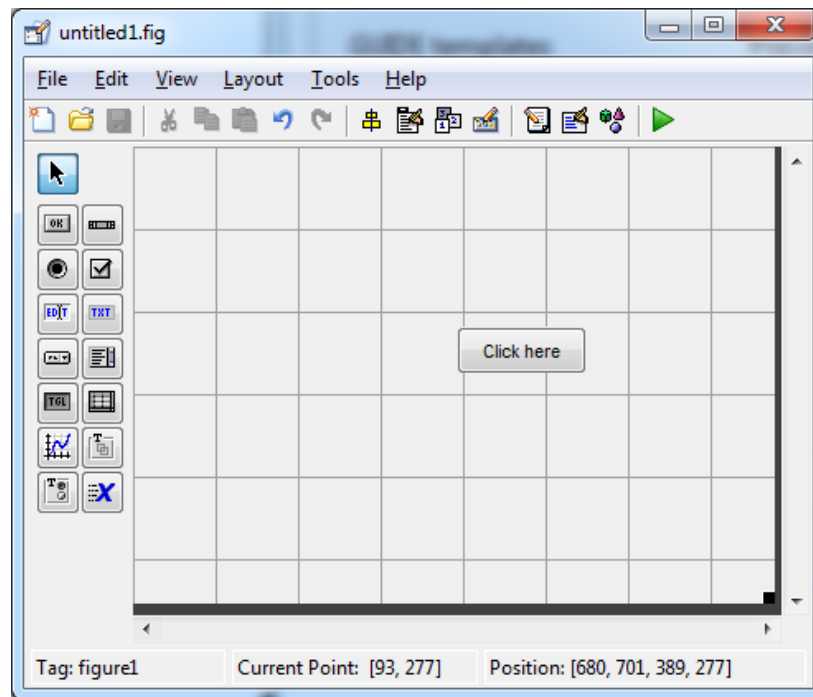
установит s в качестве обработки нажатия кнопки. Поэтому каждый раз, когда вы её нажимаете, она перемещается на новое место.

## **Редактор GUIDE**

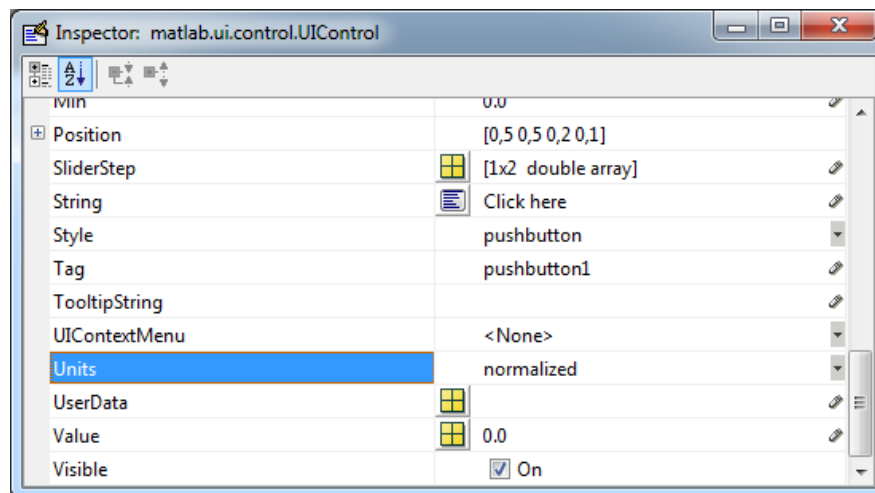
Предыдущий пример можно сделать, используя редактор пользовательских интерфейсов GUIDE. Он вызывается командой guide или в закладке меню “Home” → “New” → “Graphical User Interface”:



Выбрав Blank GUI (Default) получим следующее рабочее окно:



Слева расположены различные объекты Uicontrol, которые перетаскиванием мыши можно добавлять на рабочее поле. Сохраним проект и добавим объект pushbutton в центр рабочего поля. Кликнув 2 раза по нему, откроется окно редактора свойств:



Установим свойства 'Units' — 'normalized', 'Position' — [0.5 0.5 0.2 0.1], 'String' — 'click here'. Далее, кликнув на кнопку редактирования свойства 'Callback', попадем в текстовый редактор с автоматически созданным заголовком функции:

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)
```

Аргумент `hObject` — это идентификатор объекта, который вызвал функцию, т.е. `pushbutton1`. Аргумент `eventdata` не используется, а `handles` — структура с идентификаторами `figure` и всех объектов на ней.

Добавим в тело функции:

```
set(hObject, 'Position', [0.8*rand 0.9*rand 0.2 0.1]);
```

Запустить пользовательский интерфейс можно нажатием кнопки в виде зеленого треугольника в `GUIDE`, или в командной строке, написав название проекта.

### **Задание к лабораторной работе**

1. Создать новое графическое окно (Home → New → Graphical User Interface → Blank GUI) и сохранить его.
2. Добавить на окно объекты Axes, Push Button, Check Box. Добавить на окно объекты в соответствии с вариантом.
3. Дописать функции обработки событий так, чтобы при нажатии на кнопку Push Button график перерисовывался с новыми параметрами. Флажок Check Box показывает или скрывает сетку на графике. Функции  $y(x)$  и  $z(x)$  для построения графиков взять из соответствующего варианта Лабораторной работы 6.

Вариант	Задание
1	Выбор графика $y/z$ — Radio Button Group Цвет линий — Listbox Количество точек — Edit Text
2	Выбор графика $y/z$ — Listbox Цвет линий — Pop-up Menu Шаг между точками — Slider
3	Выбор графика $y/z$ — Pop-up Menu Цвет линий — Listbox Шаг между точками — Edit Text
4	Выбор графика $y/z$ — Radio Button Group Цвет линий — Pop-up Menu Шаг между точками — Slider
5	Выбор графика $y/z$ — Listbox Цвет линий RGB — Slider Шаг между точками — Edit Text
6	Выбор графика $y/z$ — Pop-up Menu Толщина линий — Radio Button Group $X_{\min}$ и $X_{\max}$ — Edit Text
7	Выбор графика $y/z$ — Radio Button Group Толщина линий — Slider $X_{\min}$ и $X_{\max}$ — Edit Text

8	Выбор графика y/z — Pop-up Menu Цвет линий RGB — Slider Количество точек — Edit Text
9	Выбор графика y/z — Listbox Толщина линий — Pop-up Menu Шаг между точками — Radio Button Group
10	Выбор графика y/z — Radio Button Group Цвет линий RGB — Slider Шаг между точками — Edit Text
11	Выбор графика y/z — Listbox Толщина линий — Edit Text Шаг между точками — Pop-up Menu
12	Выбор графика y/z — Pop-up Menu Толщина линий — Radio Button Group Шаг между точками — Slider
13	Выбор графика y/z — Radio Button Group Толщина линий — Edit Text Шаг между точками — Pop-up Menu
14	Выбор графика y/z — Listbox Толщина линий — Slider Шаг между точками — Edit Text
15	Выбор графика y/z — Pop-up Menu Толщина линий — Slider Шаг между точками — Listbox
16	Выбор графика y/z — Pop-up Menu Цвет линий — Listbox Количество точек — Edit Text
17	Выбор графика y/z — Listbox Цвет линий — Pop-up Menu Шаг между точками — Edit Text

18	Выбор графика y/z — Pop-up Menu Цвет линий — Slider Шаг между точками — Edit Text
19	Выбор графика y/z — Pop-up Menu Толщина линий — Radio Button Group Xmin и Xmax — Edit Text
20	Выбор графика y/z — Radio Button Group Толщина линий — Slider Количество точек — Edit Text

### **Анимация**

MATLAB предоставляет несколько способов для создания двигающейся, анимационной графики. Использование функции `drawnow` предназначено для длинной последовательности простых графиков, где изменение от кадра к кадру минимально. Ниже представлен пример, моделирующий броуновское движение. Определим количество точек

```
n = 100;
```

Температуру или скорость как

```
s = 0.002;
```

Лучшие значения этих параметров зависят от скорости вашей машины.

Сгенерируем  $n$  случайных точек с координатами  $(x,y)$  между  $-0.5$  и  $0.5$ .

```
x = rand(n,1) - 0.5;
```

```
y = rand(n,1) - 0.5;
```

Отобразим точки в квадрате со сторонами в пределах от  $-1$  до  $1$ . Сохраним вектор-идентификатор для точек и установим свойство `MarkerSize` равным  $18$ . В бесконечном цикле будем пересчитывать координаты точек и с помощью функции `drawnow` перерисовывать график.

```
h = plot(x , y , ' . ' );
```

```
axis([-1 1 -1 1])
```

```
axis square
```

```
grid off
```

```
set(h , 'MarkerSize' , 18 );
```

```
while 1
```

```
    x = x + s*randn(n,1);
```

```
    y = y + s*randn(n,1);
```

```
    set(h, 'Xdata' , x , 'Ydata' , y);
```

```
    drawnow update
```

```
end
```

Прервать выполнение вычислений можно комбинацией клавиш Ctrl+C.

## **Movie**

Если вы увеличиваете число точек в броуновском движении, например,  $n = 300$ , то в этом случае движение не будет очень подвижным, так как потребуется слишком много времени на каждом шаге. Становится более эффективным сохранить определенное число кадров как bitmap и проигрывать их как кино.

Во-первых, пусть число кадров

```
nframes = 50;
```

Во-вторых, установим первый график, как и ранее, но с отключенными осями:

```
x = rand(n,1) - 0.5;  
y = rand(n,1) - 0.5;  
h = plot(x, y, ' . ');  
set(h, 'MarkerSize' , 18);  
axis([-1 1 -1 1]);  
axis square  
axis off  
grid off
```

Теперь создадим специальную структуру для сохранения нашего фильма:

```
M = moviein(nframes);
```

Сгенерируем полученный фильм и, используя getframe, сохраним каждый кадр:

```
for k = 1:nframes  
    x = x + s*randn(n,1);  
    y = y + s*randn(n,1);  
    set(h, 'XData' , x , 'YData' , y)  
    M( : , k) = getframe;  
end
```

Это установит большую матрицу с nframes столбцами. Каждый столбец достаточно длинный, чтобы сохранить один кадр. Полное количество требуемой памяти пропорционально количеству кадров и области текущих осей, но она не зависит от сложности отдельных графиков. Для 50 кадров нашего примера необходимо около 37 мегабайт памяти. Узнать это можно функцией whos( 'M' ).

Теперь проиграем фильм 3 раза:

```
movie(M, 3)
```

## **Лабораторная работа 11. Математическая статистика.**

### **Законы распределения случайной величины в MATLAB**

В MATLAB имеются функции для задания случайных величин, расчета плотностей вероятности и функций распределения для многих известных законов распределения. Далее приведены сокращенные названия некоторых законов распределения:

- 'beta' — Beta distribution;
- 'bin' — Binomial distribution;
- 'chi2' — Chi-square distribution;
- 'exp' — Exponential distribution;
- 'ev' — Extreme value distribution;
- 'f' —  $F$  distribution;
- 'gam' — Gamma distribution;
- 'gev' — Generalized extreme value distribution;
- 'gp' — Generalized Pareto distribution;
- 'geo' — Geometric distribution;
- 'hyge' — Hypergeometric distribution;
- 'logn' — Lognormal distribution;
- 'nbin' — Negative binomial distribution;
- 'ncf' — Noncentral  $F$  distribution;
- 'nct' — Noncentral  $t$  distribution;
- 'ncx2' — Noncentral chi-square distribution;
- 'norm' — Normal distribution;
- 'poiss' — Poisson distribution;
- 'rayl' — Rayleigh distribution;
- 't' —  $t$  distribution;
- 'unif' — Uniform distribution;
- 'unid' — Discrete uniform distribution;
- 'wbl' — Weibull distribution.

Для моделирования случайных чисел из заданного закона распределения используются функции, состоящие из сокращенного названия закона распределения и окончания rnd. Например:

$y = \text{unifrnd}(a, b)$  — возвращает случайное число из равномерного на промежутке  $(a ; b)$  распределения.

$y = \text{normrnd}(m, \text{sigma})$  — возвращает случайное число из нормального распределения, где  $m$  — математическое ожидание,  $\text{sigma}$  — среднее квадратическое отклонение.

$y = \text{chi2rnd}(k)$  — возвращает случайное число из распределения  $\chi^2$  с  $k$  степенями свободы.



Для создания матрицы случайных величин можно в качестве дополнительных параметров таких функций указать нужные размерности. Например,

$M = \text{trnd}(k, m, n)$  — возвращает матрицу размерности  $m$  на  $n$  случайных чисел из распределения Стьюдента с  $k$  степенями свободы.

$V = \text{frnd}(k1, k2, m, n)$  — возвращает вектор из  $n$  случайных чисел из распределения Фишера с  $k1$  и  $k2$  степенями свободы.

### **Функции для расчета плотности вероятности**

Имена функций для расчета плотностей вероятности оканчиваются буквами pdf (probability density function). Например,

$y = \text{unifpdf}(x, a, b)$  — расчет значения плотности вероятности в точке  $x$  для равномерного на промежутке  $(a; b)$  распределения.

$y = \text{normpdf}(x, m, \text{sigma})$  — расчет значения плотности вероятности в точке  $x$  для нормального распределения, где  $m$  — математическое ожидание,  $\text{sigma}$  — среднее квадратическое отклонение.

$y = \text{expnpdf}(x, \text{lambda})$  — расчет значения плотности вероятности в точке  $x$  для экспоненциального распределения с параметром  $\text{lambda}$ , равным математическому ожиданию случайной величины.

### **Функции MATLAB для расчета функций распределения случайной величины**

Имена функций MATLAB для расчета функций распределения случайной величины оканчиваются буквами cdf (cumulative distribution function). Например,

$y = \text{chi2cdf}(x, k)$  — расчет значения функции распределения в точке  $x$  для распределения  $\chi^2$  с  $k$  степенями свободы.

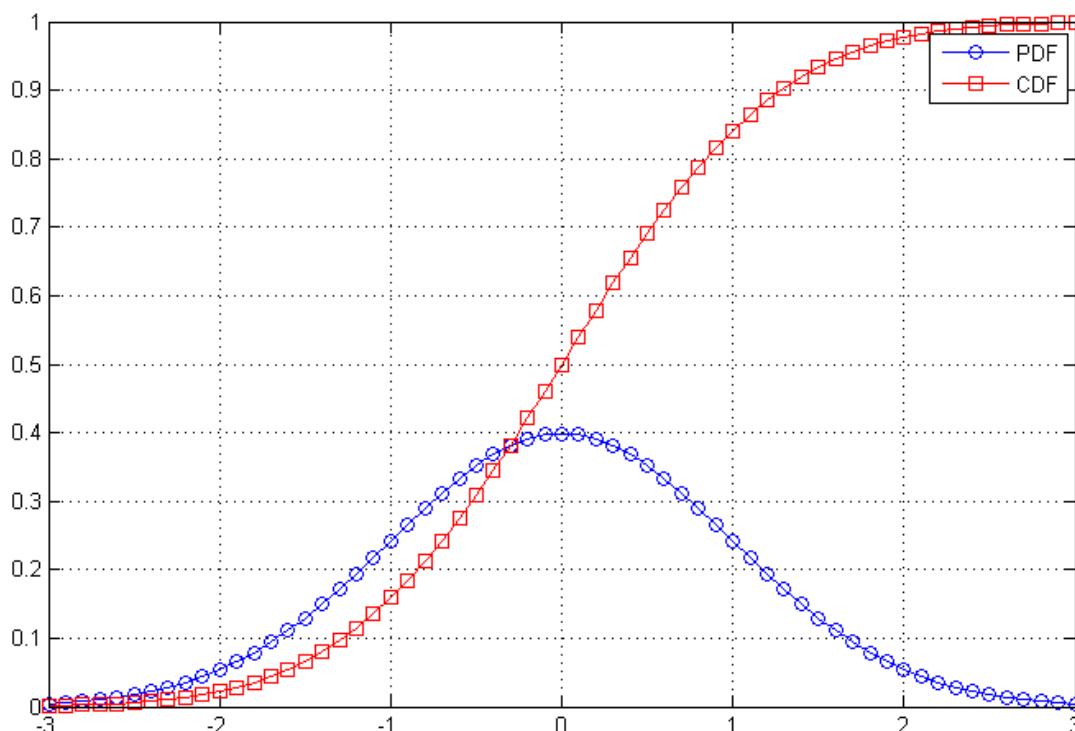
$y = \text{tcdf}(x, k)$  — расчет значения функции распределения в точке  $x$  для распределения Стьюдента с  $k$  степенями свободы.

$y = \text{fcdf}(x, k1, k2)$  — расчет значения функции распределения в точке  $x$  для распределения Фишера с  $k1$  и  $k2$  степенями свободы.

Пример расчета плотности вероятности и функции распределения случайной величины с нормальным законом распределения:

```
mu = 0;  
sigma = 1;  
X = -3:0.1:3;  
P = normpdf(X, mu, sigma);  
C = normcdf(X, mu, sigma);
```

```
figure(1); clf;
plot(X, P, '-ob', X, C, '-sr');
legend('PDF', 'CDF');
grid on;
```



### **Функции вычисления статистических параметров**

Функция `max(X)` возвращает наибольший элемент вектора  $X$ , или вектор-строку, состоящую из максимальных элементов каждого столбца матрицы  $X$ . Вторым возвращаемым аргументом является индекс наибольшего элемента.

Функция `min(X)` возвращает наименьший элемент вектора  $X$ , или вектор-строку, состоящую из минимальных элементов каждого столбца матрицы  $X$ . Вторым возвращаемым аргументом является индекс наименьшего элемента.

Функция `mean(X)` возвращает среднее арифметическое значений элементов вектора  $X$  или вектор-строку, состоящую из средних арифметических значений элементов каждого столбца матрицы  $X$ .

Функция `median(X)` возвращает медиану вектора  $X$ .

Функция `mode(X)` возвращает моду вектора  $X$ .

Функция `std` в форме `std(X)` или `std(X,0)` возвращает несмещенную оценку среднеквадратического отклонения вектора  $X$ . В форме `std(X,1)` возвращает смещенную оценку среднеквадратического отклонения вектора  $X$ .

Функция `var` в форме `var(X)` или `var(X,0)` возвращает несмещенную оценку дисперсии вектора  $X$ . В форме `var(X,1)` возвращает смещенную оценку дисперсии вектора  $X$ .

## Гистограмма

Функция `hist` используется для получения и отображения гистограммы, а также для получения данных для гистограммы. Для получения данных для гистограммы функция `hist` записывается в следующем виде:

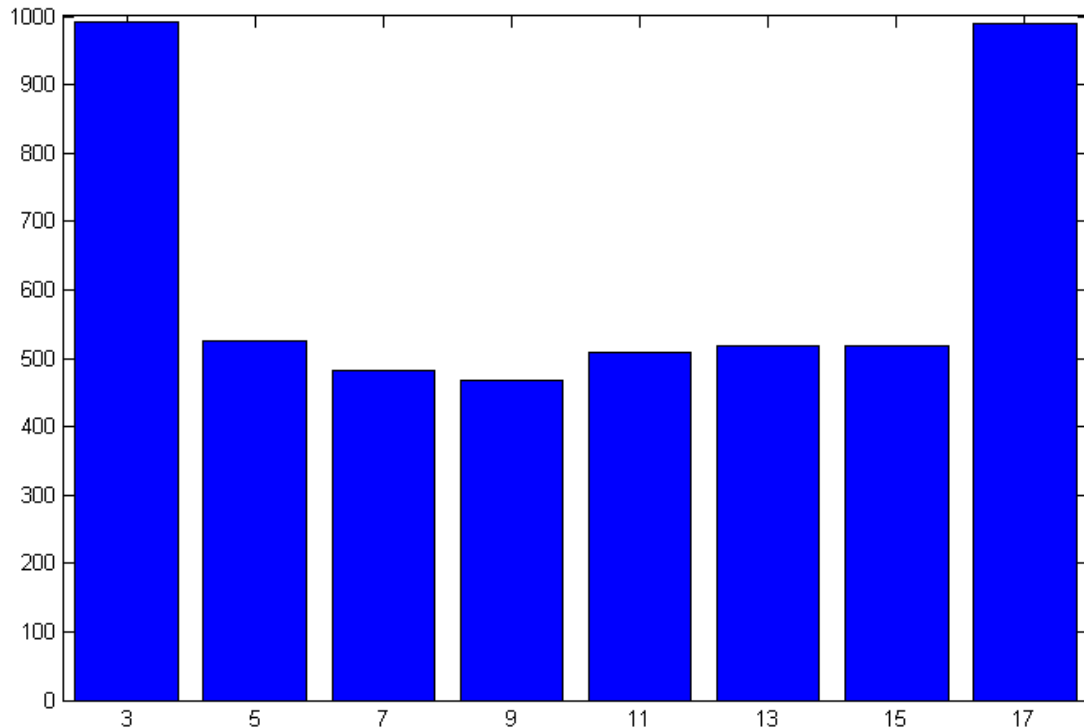
$N = \text{hist}(X, Y)$ , где  $Y$  — вектор, возвращает количество элементов вектора  $X$ , попавших в интервалы с центрами, заданными вектором  $Y$ . Число интервалов в этом случае равно числу элементов вектора  $Y$ ;

$[N, Y] = \text{hist}(X, M)$ , возвращает вектор  $N$  чисел попаданий вектора  $X$  в  $M$  интервалов группировки одинаковой длины и данные о центрах интервалов  $Y$  (если  $X$  — матрица, то `hist` работает со столбцами).

Для отображения гистограммы в виде столбцовой диаграммы используется функция `bar(Y, N, 'цвет_стиль_маркер')`.

Пример вычисления и построения гистограммы:

```
X = randi(20,1,5000);  
Y = [3 5 7 9 11 13 15 17];  
N = hist(X,Y);  
figure(1); clf;  
bar(Y,N,'b');
```



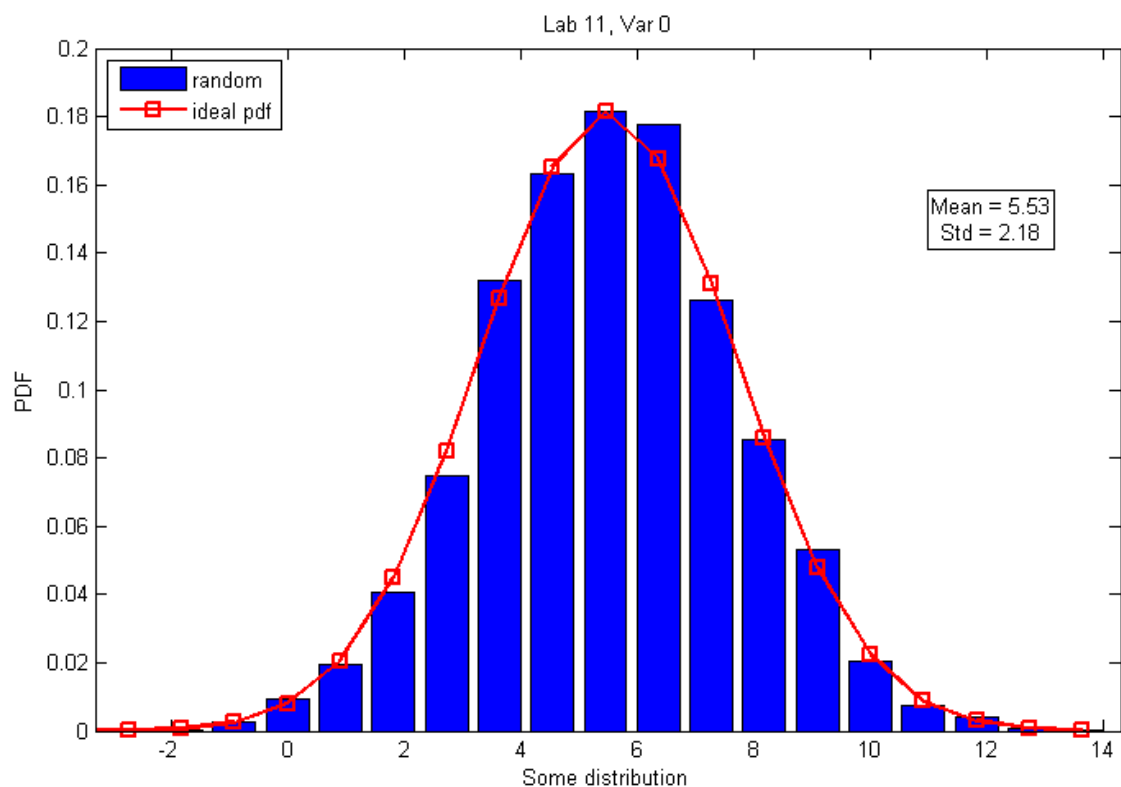
### Задание к лабораторной работе

1. Сгенерировать массив из 5000 случайных чисел с заданными законом распределения и параметрами. Вычислить среднее математическое значение и среднеквадратическое отклонение этих случайных чисел.
2. Исходя из максимального и минимального значений полученного массива, сформировать вектор точек для вычисления гистограммы. В нем должно быть около 20 точек.
3. С помощью функции `hist` вычислить уровни гистограммы по полученному вектору точек. Произвести нормировку уровней гистограммы так, чтобы ее площадь стала равна 1.
4. Вычислить значения плотности распределения вероятности (pdf) заданного закона в тех же точках, что вычислялась гистограмма.
5. На одном графике изобразить гистограмму и плотность распределения вероятностей разными цветами. Добавить легенду.
6. Подписать оси графика, в которых указать закон распределения вероятности.
7. Добавить заголовок графика с указанием номера варианта.
8. В произвольное место на графике добавить текстовое поле с белым фоном и черной рамкой, в котором вывести среднее математическое значение и среднеквадратическое отклонение, вычисленные ранее.

Вариант	Задание
1	Normal distribution с параметрами 1.5 и 1.2
2	Beta distribution с параметрами $A = 0.7$ и $B = 0.7$
3	Chi-square distribution с параметром $V = 6$
4	Exponential distribution с параметром 2
5	Extreme value distribution с параметрами 9 и 1.1
6	Lognormal distribution с параметрами 1.5 и 0.4
7	F distribution с параметрами $V1 = 15$ и $V2 = 30$
8	Gamma distribution с параметрами $A = 5$ и $B = 1.2$
9	Binomial distribution с параметрами $N = 400$ и $P = 0.02$
10	Geometric distribution с параметром $p = 0.3$

11	Poisson distribution с параметром 9
12	Student's t distribution с параметром $\nu = 4$
13	Uniform distribution с параметрами $A = -0.7$ и $B = 0.7$
14	Weibull distribution с параметрами $A = 1.7$ и $B = 3.7$
15	Rayleigh distribution с параметром $B = 4$
16	Noncentral T distribution с параметрами $V = 20$ и $\Delta = 1$
17	Hypergeometric distribution с параметрами $M = 200$ , $K = 40$ и $N = 50$
18	Noncentral chi-square distribution с параметрами $V = 1$ и $\Delta = 20$
19	Noncentral F distribution с параметрами $\nu_1 = 40$ , $\nu_2 = 90$ и $\Delta = 8$
20	Generalized extreme value distribution с параметрами $k = 0.1$ , $\sigma = 0.1$ и $\mu = 4$

Пример выполнения лабораторной работы:

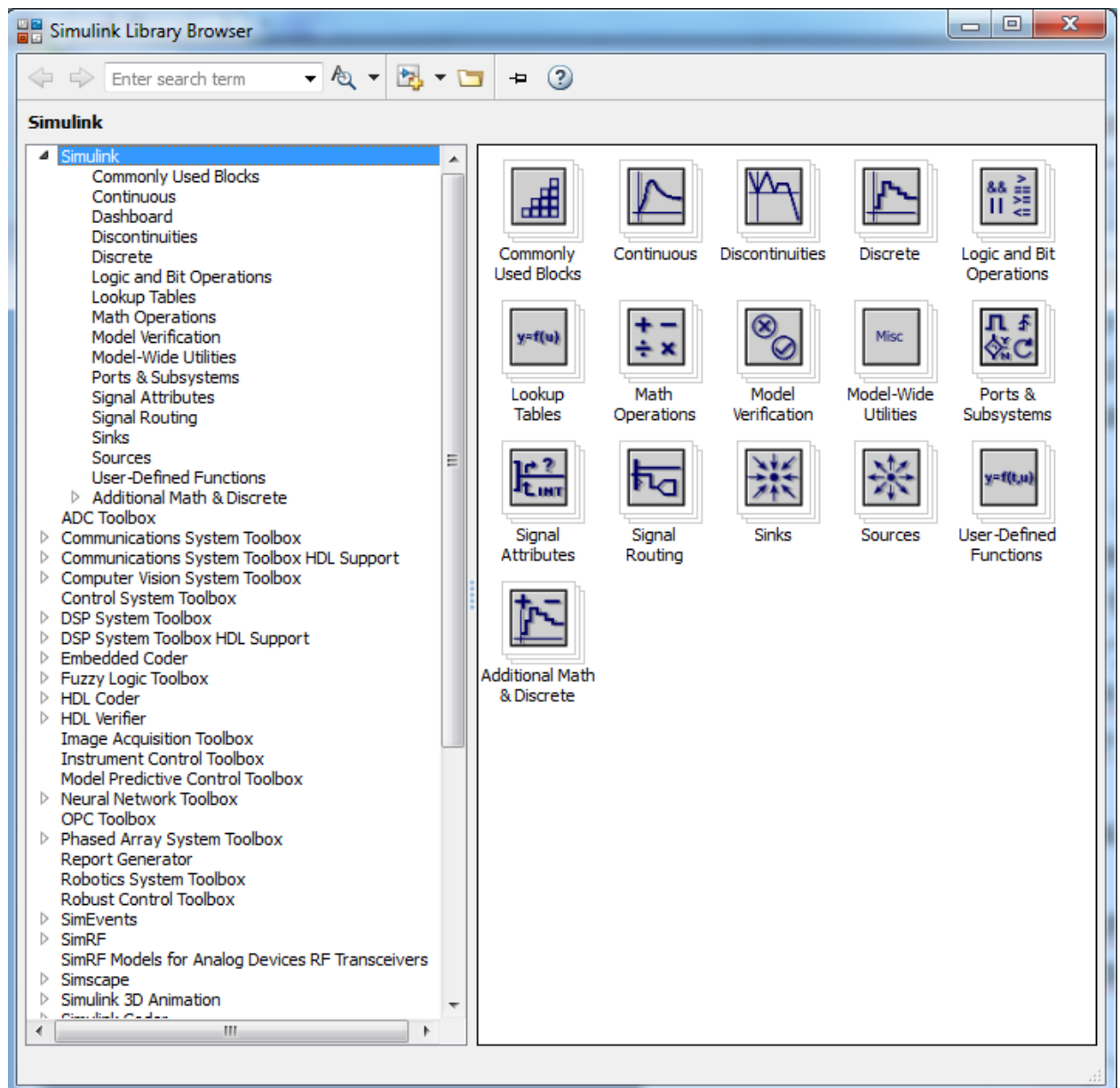


## Лабораторная работа 12. Основы работы в SIMULINK.

### Запуск пакета SIMULINK

Запуск пакета SIMULINK осуществляется одним из следующих способов:

- с помощью кнопки на панели инструментов (при этом вызывается окно браузера, называемое также «окно обозревателя библиотеки» — Simulink Library Browser);
- набором в строке командного окна слова `simulink` (также вызывается окно браузера);
- последовательным выбором пунктов меню “Home” → “New” → “Simulink Model” (открывается окно для создания модели);
- с помощью кнопки открытия документа на панели инструментов (вызывается окно с построенной ранее моделью, сохраненной в виде `slx` или `mdl`-файла).

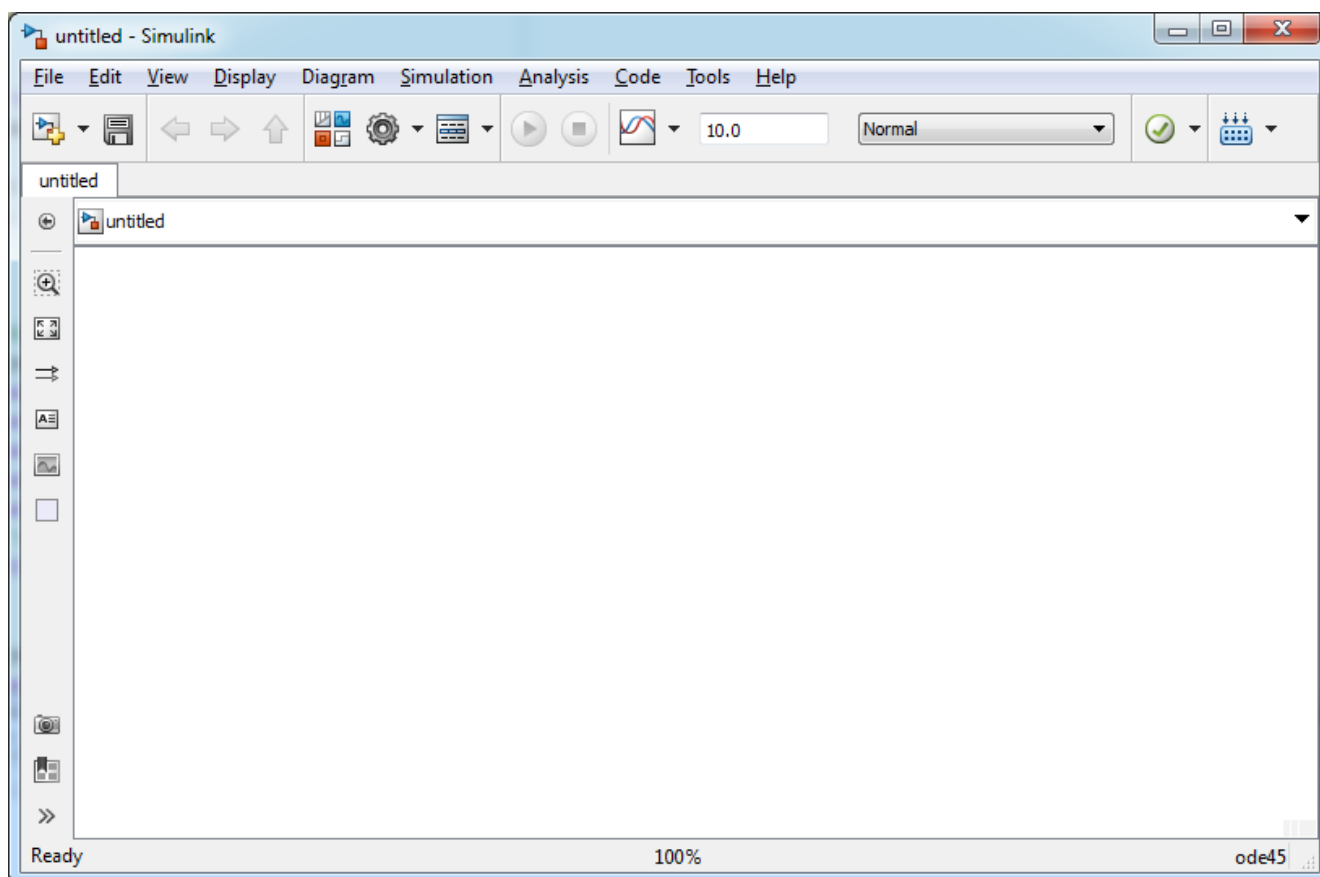


При вызове окна браузера автоматически открывается раздел библиотеки SIMULINK в левой (подменю в виде дерева) и правой (пиктограммы подразделов) частях окна.

### **Окно для создания модели**

Окно для создания модели содержит следующие области:

- панель названия окна или имени модели;
- панель меню;
- панель инструментов;
- окно для непосредственного создания модели;
- панель масштабирования и визуального оформления (вертикальная слева);
- строка состояния, содержащая сведения о состоянии модели (снизу окна).



Вновь открытое окно модели имеет имя *untitled* — «Без названия» с соответствующим номером, если открыты несколько окон модели. При сохранении созданной модели в виде slx-файла через пункты меню “File” → “Save as...” открывается окно сохранения в текущую рабочую папку системы MATLAB. Название должно начинаться с букв (использовать только латинский шрифт)

и содержать при необходимости цифры. В качестве разделителя допускается использовать только черту подчеркивания. Пример названия: *Model\_3\_04*. После того, как выполнено сохранение, название модели автоматически присваивается в качестве названия окну.

При работе с моделью целесообразно пользоваться кнопками панели инструментов, основные из которых:

Run — запуск процесса моделирование-пауза-продолжение;

Stop — закончить моделирование. Кнопка становится доступной после начала моделирования.

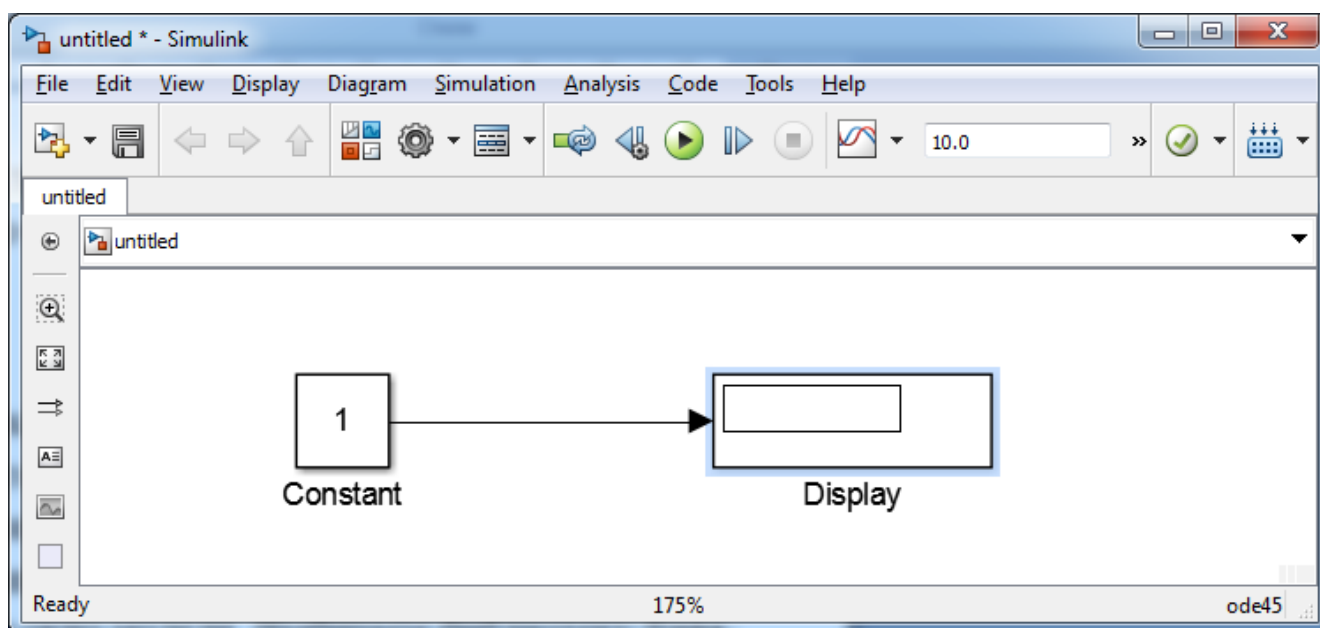
Library Browser — открыть окно браузера — обозревателя библиотеки блоков.

Model Configuration Parameters — открыть дополнительное окно обозревателя модели.

Up to Parent — переход в подсистему высшего уровня иерархии. Команда доступна только из подсистемы низшего уровня.

## **Построение модели**

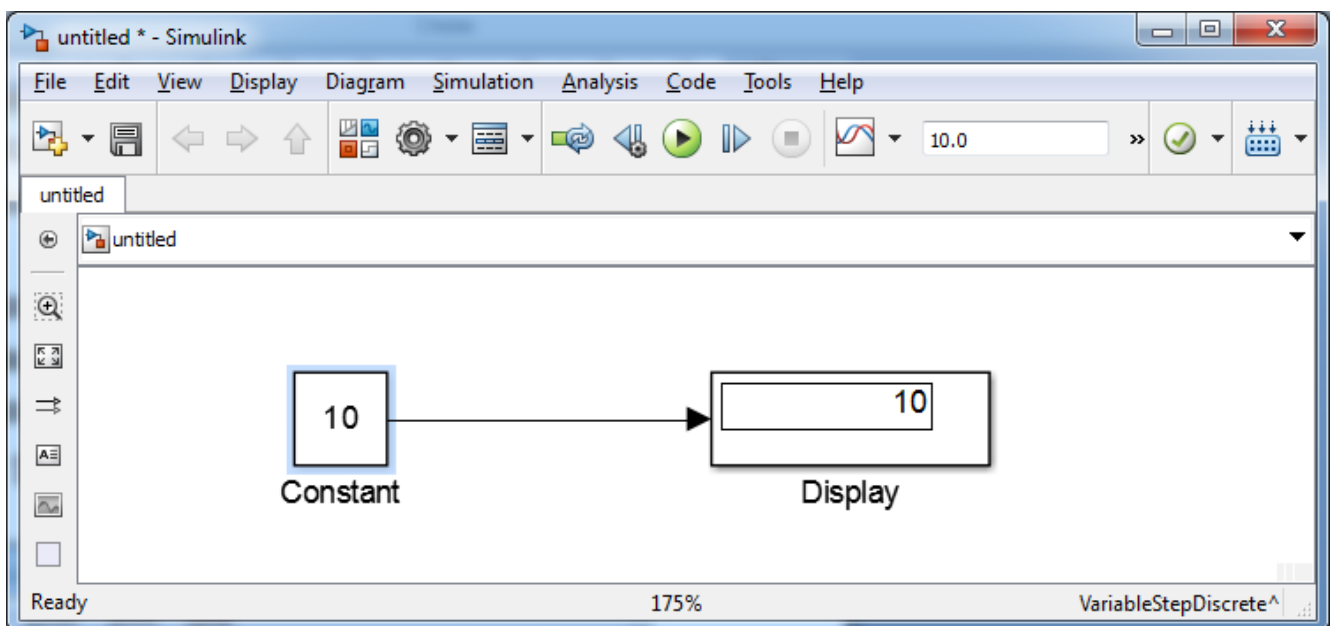
Соберем простейшую схему — источник постоянного напряжения 10 В и измеритель этого напряжения. Для этого в разделе SIMULINK щелкнем в левой части на строчке Source дерева. Пиктограммы разделов заменятся на пиктограммы блоков раздела Source. Найдем пиктограмму блока Constant (Источник постоянного сигнала), и перетянем его в окно модели. Выбираем библиотеку Sinks и найдем измерительный блок Display (Цифровой вольтметр). Перетащим его в окно модели. Соединим оба блока, чтобы получилась схема измерения:





Теперь необходимо настроить блоки. У блока Constant выставляется выходное напряжение 10 В. При двойном щелчке мыши по пиктограмме этого блока открывается окно настройки Source Block parameters: Constant. В текстовом окне Constant value надо исправить 1 на 10. Блок Display можно не настраивать, хотя настройки у него также имеются.

Для запуска построенной модели следует нажать кнопку Run. Процесс моделирования в такой задаче выполняется относительно быстро, что видно по изменениям надписей в строке состояния окна. По окончании на экране блока Display появляется результат измерения, равный 10:



Порядок действий при работе с моделью в SIMULINK следующий:

- вызвать окно браузера и окно модели;
- переместить необходимые блоки из библиотеки SIMULINK в окно модели и расположить их так, чтобы было удобно соединять между собой;
- произвести необходимые соединения блоков;
- выполнить настройку блоков, вызвав окно параметров у каждого блока двойным щелчком мыши;
- при необходимости сохранить модель в качестве slx-файла (в более ранних версиях использовалось расширение mdl);
- запустить моделирование;
- закрыть по очереди все окна и библиотеки SIMULINK, а при необходимости выйти из MATLAB.

## **Основные разделы библиотеки SIMULINK**

Основные разделы библиотеки SIMULINK:

- Continuous — блоки аналоговых (непрерывных) сигналов;
- Discontinuous — блоки нелинейных элементов;
- Discrete — блоки дискретных (цифровых) сигналов;
- Logic and Bit Operations — блоки логических операций;
- Look-Up Tables — блоки для формирования таблиц;
- Math Operations — блоки для реализации математических операций;
- Model Verification — блоки для проверки параметров сигналов;
- Model-Wide Utilities — подраздел дополнительных утилит;
- Ports & Subsystems — порты и подсистемы;
- Signal Attributes — блоки для изменения параметров сигналов;
- Signal Routing — блоки, определяющие маршруты сигналов;
- Sinks — блоки для измерения, контроля и сохранения сигналов;
- Sources — источники сигналов;
- User-Defined Function — функции, задаваемые пользователем.

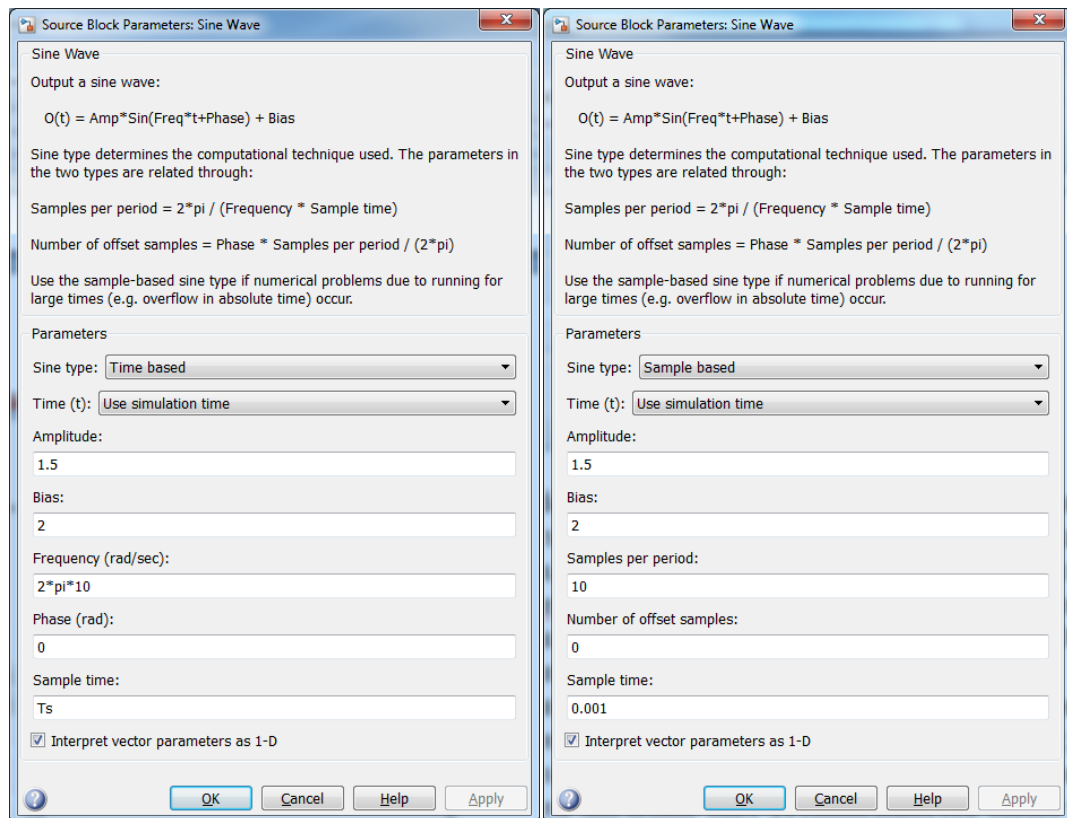
Рассмотрим основные блоки библиотеки SIMULINK, которые используются чаще всего.

### **Генератор синусоидального сигнала Sine Wave**

Блок позволяет создать сигнал синусоидальной формы с заданной частотой, амплитудой, фазой и смещением.

Формирование сигнала осуществляется в соответствии с алгоритмом, выбираемым в текстовом окне Sine type: Time based или Sample based. В первом случае сигнал формируется по текущему времени для непрерывных систем, а во втором случае формирование сигнала осуществляется по количеству тактов в периоде.

Окна параметров блока, вызываемые двойным щелчком левой клавиши мыши по пиктограмме генератора, для двух вариантов формирования сигнала показаны ниже.

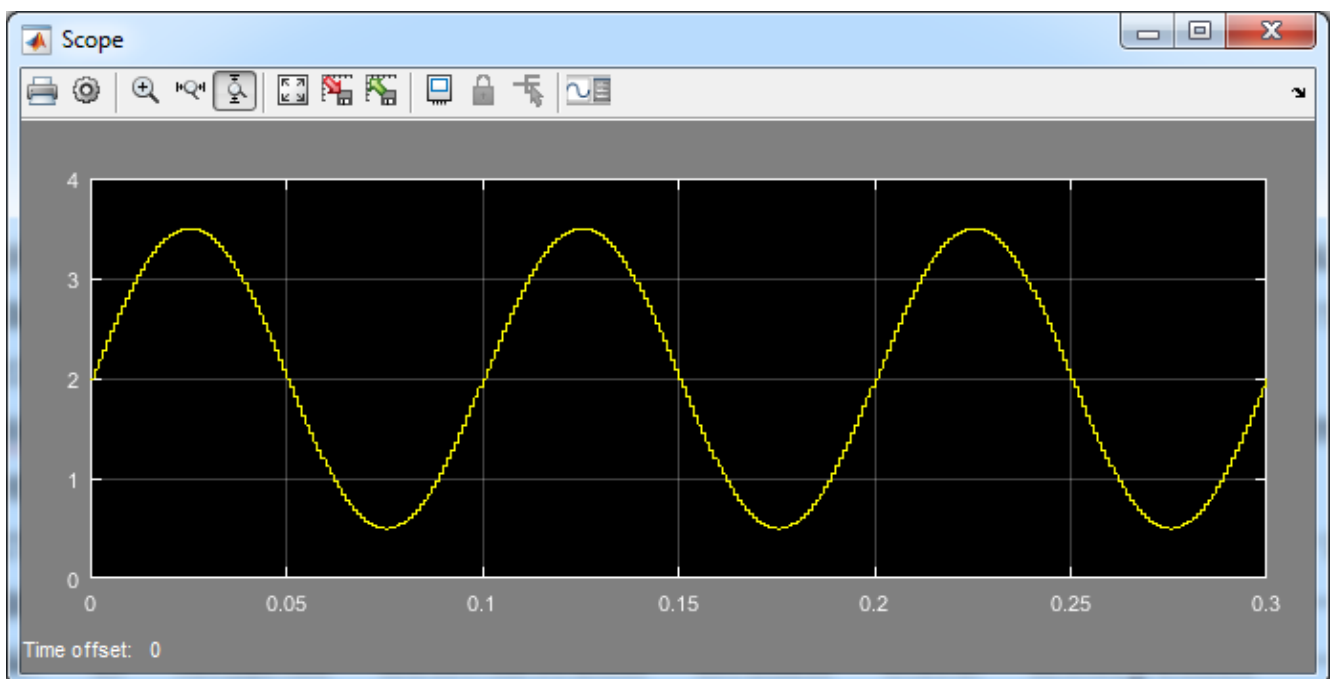
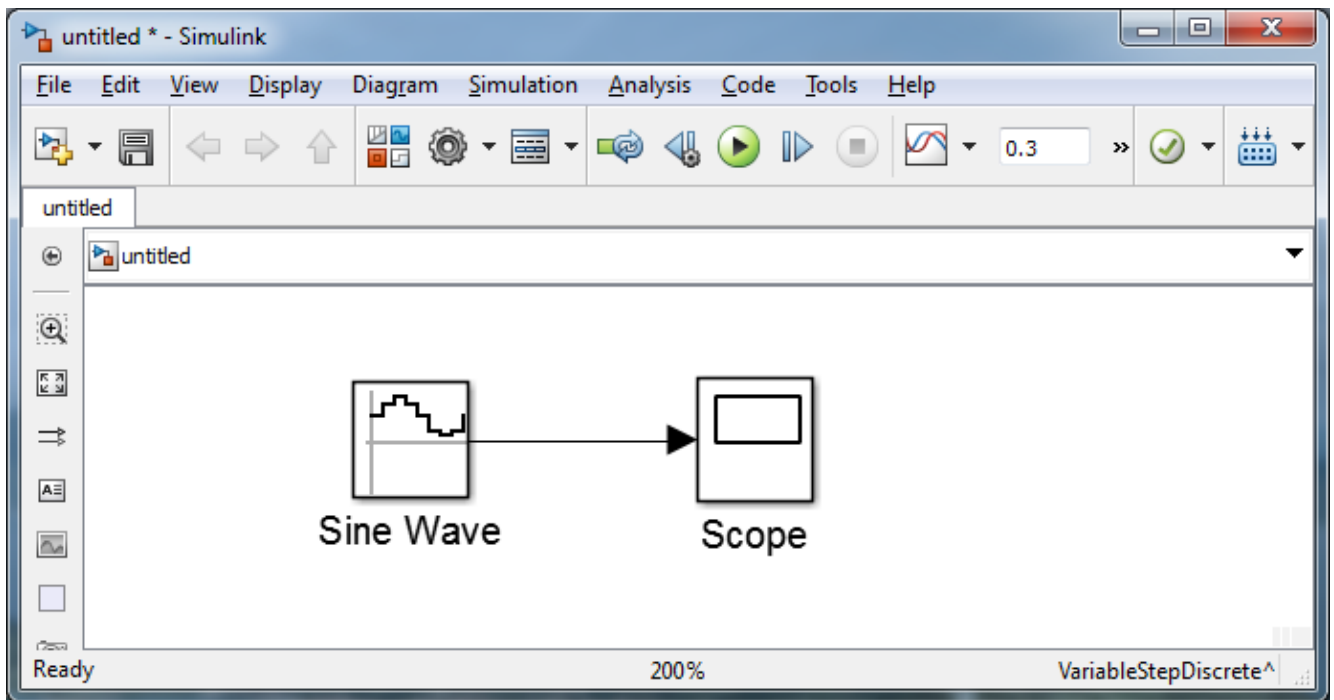


Параметры блока: Amplitude — амплитуда сигнала, Bias — постоянная составляющая (смещение) в сигнале, Frequency (rad/sec) — частота (рад/с), Phase (rad) — начальная фаза (рад), Samples per period — количество тактов за период, Number of offset samples — начальная фаза в тактах, Sample time — период тактового сигнала, Interpret vector parameters as 1-D — интерпретация вектора как массива скаляров.

Параметры блоков можно задавать числами или переменными, предварительно объявленными в рабочей области MATLAB или в функции инициализации модели.

Период тактового сигнала принимает значение 0 (по умолчанию используется для моделирования непрерывных систем); положительное значение задается при моделировании дискретных систем; -1 (период тактового сигнала устанавливается таким же, как и у предшествующего блока).

В качестве примера приведена тестовая модель и результат моделирования в виде временной диаграммы — осциллограммы. В модели использован новый блок измерения — осциллограф (Scope), который будет рассмотрен далее.

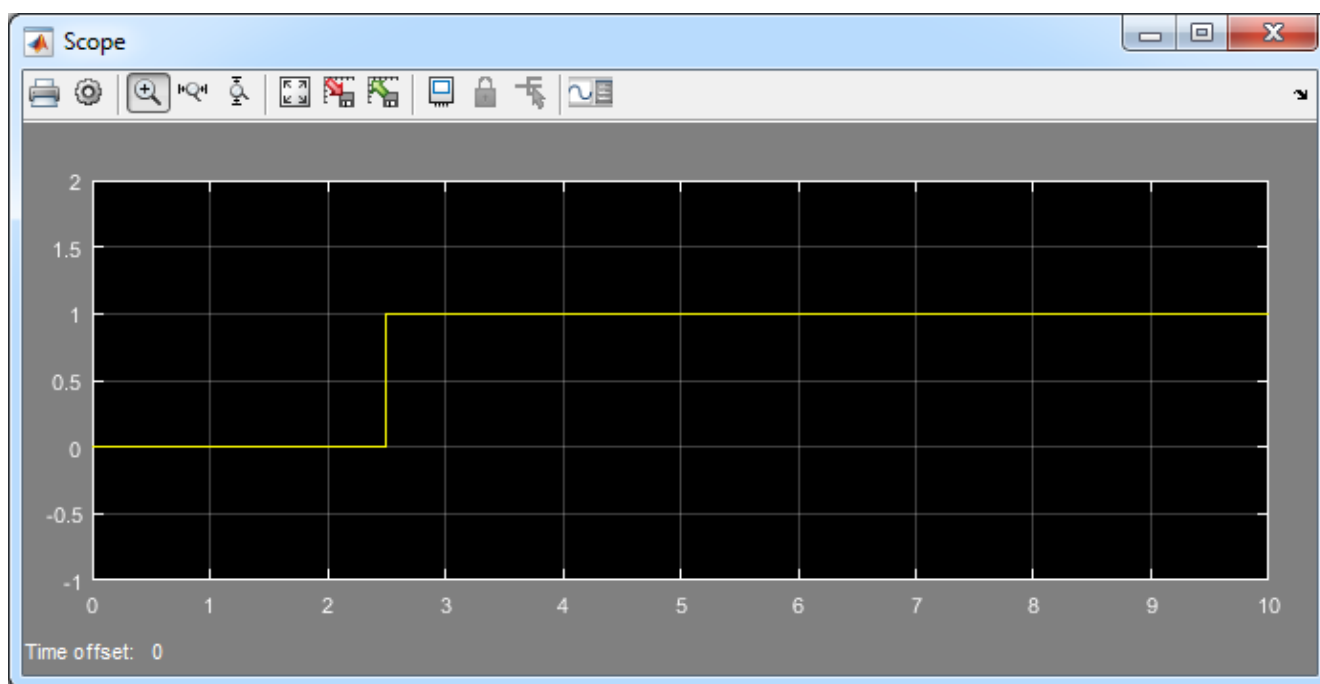
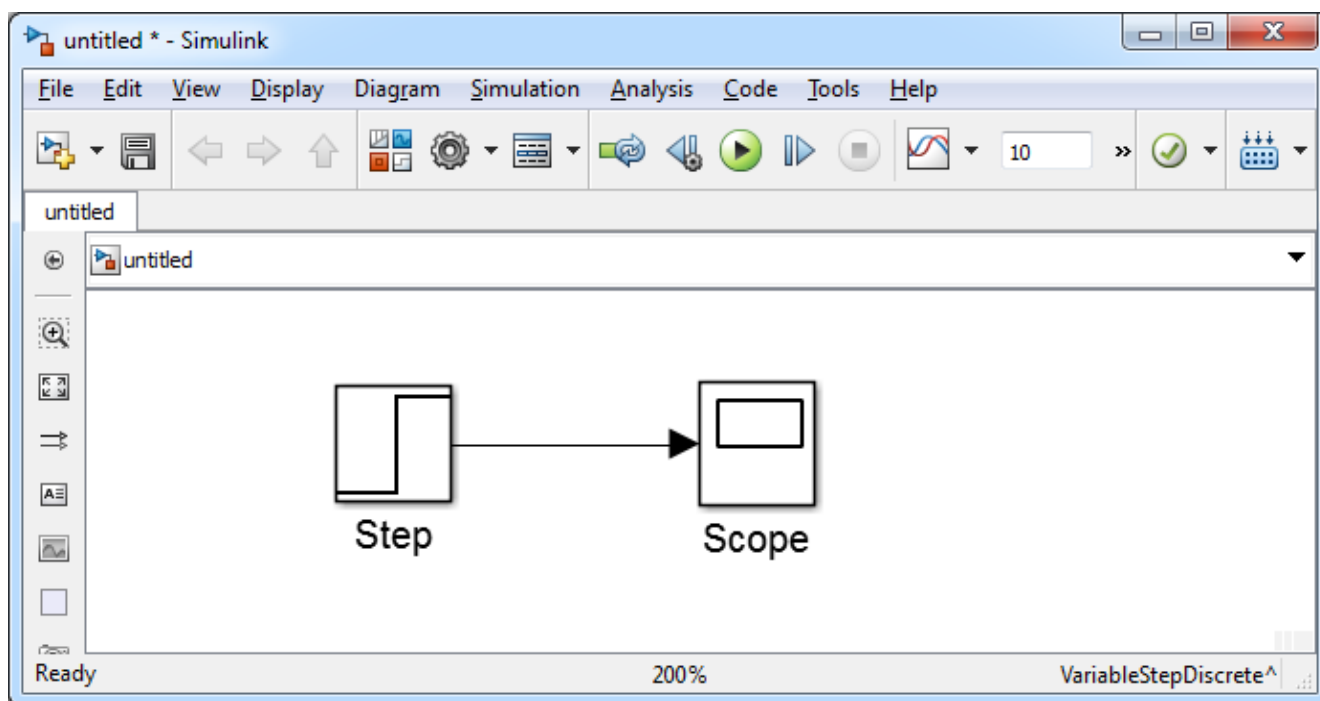


### **Генератор ступенчатого сигнала Step**

Блок позволяет создать сигнал в виде единичного скачка.

Параметры блока: Step time — время начала перепада сигнала, Initial value — начальное значение сигнала, Final value — конечное значение сигнала (оба значения могут быть положительными или отрицательными), Sample time — период тактового сигнала, Interpret vector parameters as 1-D — интерпретация вектора как массива скаляров, Enable zero crossing detection — определение прохождения сигнала через нуль.

Тестовая модель с генератором ступенчатого сигнала и осциллографом (Scope), а также результат моделирования приведены на следующих двух рисунках:



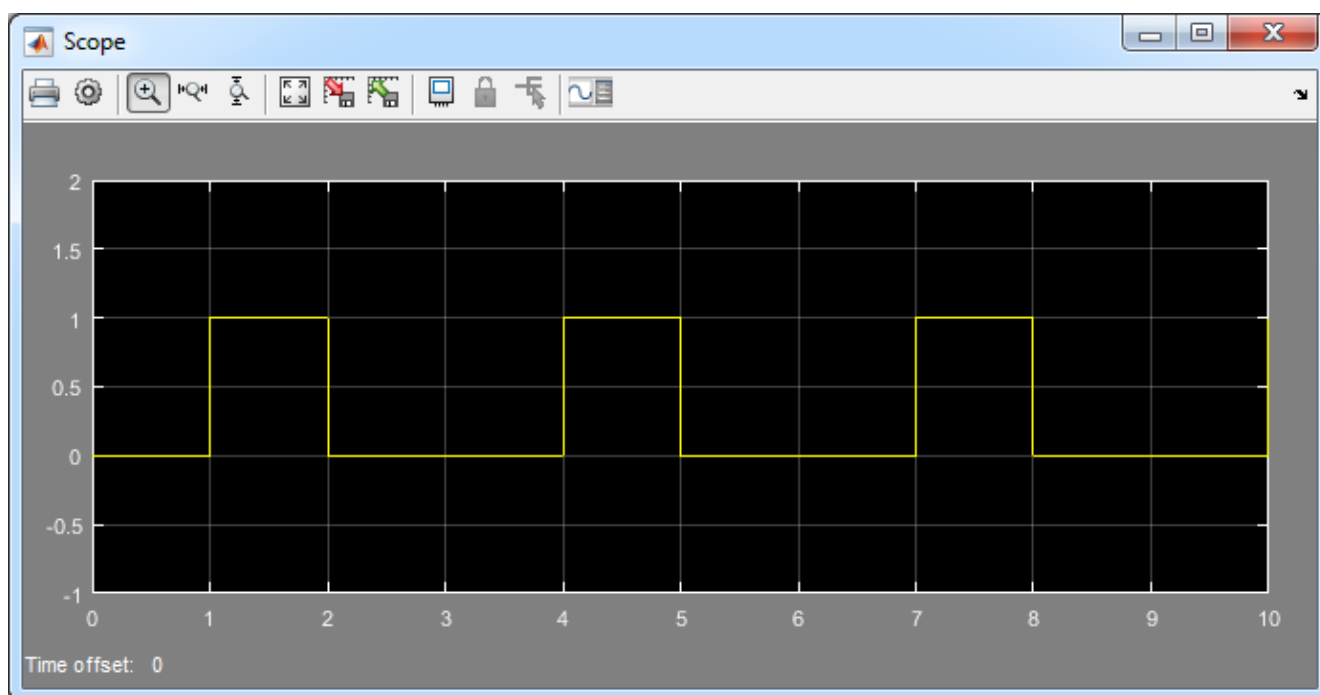
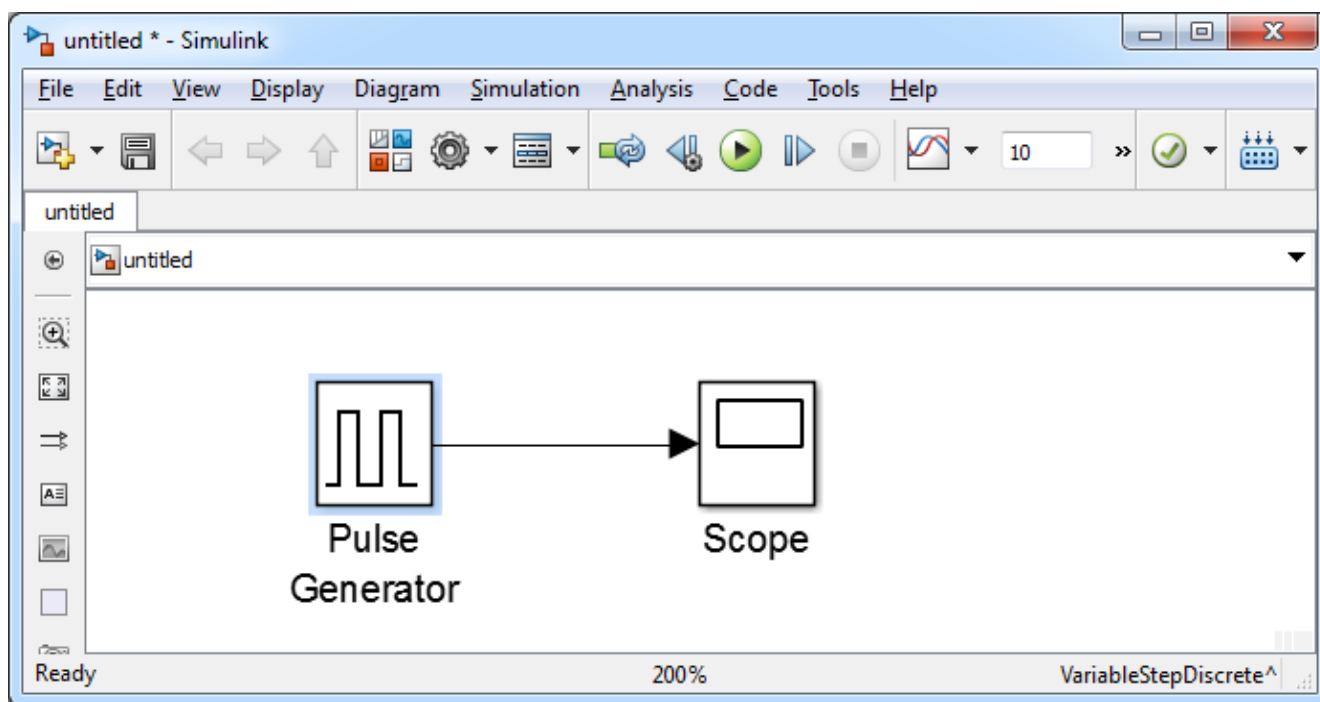
### **Генератор импульсного сигнала Pulse Generator**

Блок предназначен для формирования импульсного напряжения.

Параметры блока: Pulse Type — способ формирования сигнала: Time based — по текущему времени и Sample based — по тактовому сигналу;

Amplitude — амплитуда, Period — период (задается в секундах или количеством тактов), Pulse width — ширина импульса (задается в процентах по отношению к периоду или количеством тактов), Phase delay — фазовая задержка (задается в секундах или количеством тактов), Sample time — период тактового сигнала, Interpret vector parameters as 1-D — интерпретация вектора как массива скаляров.

Схема с использованием Pulse Generator и результаты моделирования приводятся далее.



## Осциллограф Scope

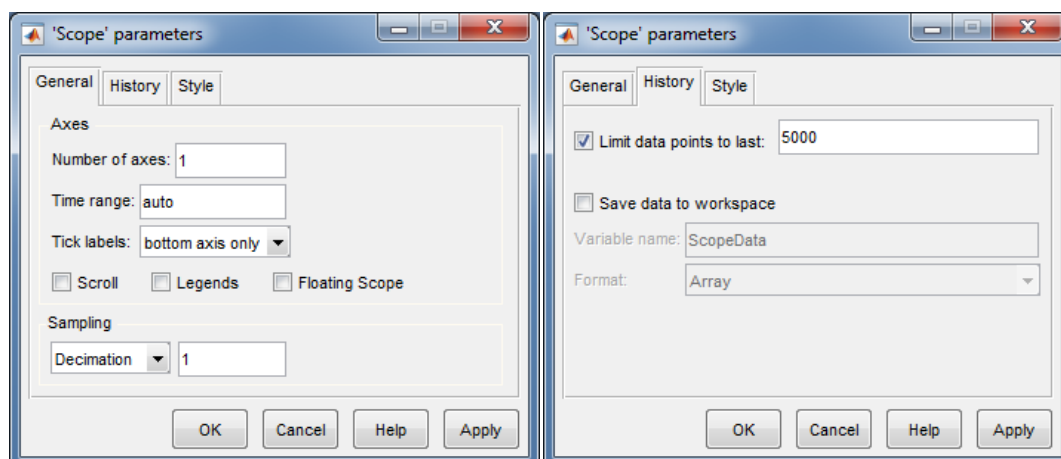
Блок предназначен для построения временных диаграмм сигналов.

Дает возможность наблюдения за ходом процессов при моделировании. Окно для наблюдения за сигналами (окно осциллограммы) открывается двойным щелчком мыши по пиктограмме Scope, что выполняется на любой фазе моделирования. Для настройки этого окна используются кнопки на его панели инструментов:

- Print — печать содержимого окна осциллограмм;
- Parameters — вызов окна настройки параметров;
- Zoom — увеличение масштаба по обеим осям;
- Zoom X-axis — увеличение масштаба по горизонтальной оси;
- Zoom Y-axis — увеличение масштаба по вертикальной оси;
- Autoscale — автоматическая установка масштаба по обеим осям;
- Save current axes settings — сохранение текущих настроек окна;
- Restore saved axes settings — установка ранее сохраненных настроек окна;
- Floating scope — перевод осциллографа в изменяющийся режим;
- Lock/Unlock axes selection — закрепить/разорвать связь между текущей координатной системой окна и отображаемым сигналом (при включенном режиме Floating scope);
- Signal selections — выбор сигнала для отображения (при включенном режиме Floating scope);
- Try TimeScope — переключение в тестовый режим TimeScope.

Изменять масштаб осциллограммы можно через команду Axes properties... (свойства осей), которая выбирается из контекстного меню.

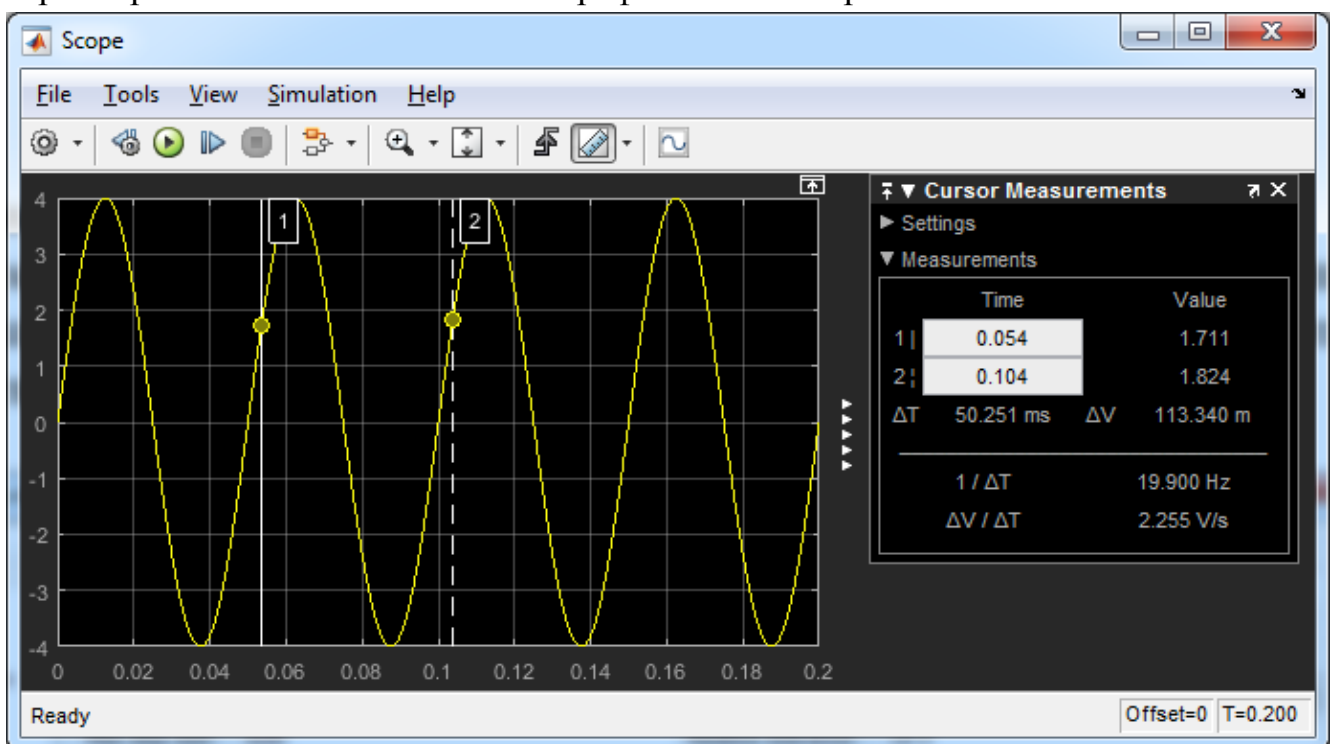
Для настройки осциллографа используется окно задания параметров 'Scope' parameters, которое вызывается кнопкой Parameters и содержит три закладки: General — общие параметры, History — параметры сохранения сигнала в памяти системы MATLAB и Style — параметры графического оформления.



На закладке General задаются Number of axes (число осей или число входов осциллографа); Time range (величина временного интервала, на котором отображается график — модельное время); Tick labels — вывод и скрытие осей и меток: all — подписи для всех осей; none — без осей и подписей к ним; bottom axis only — подписи горизонтальной оси только для нижнего графика. Sampling — установка параметров вывода графиков в окне: Decimation — прореживание или кратность вывода точек графика (так, при кратности 2 выводится каждая вторая точка); Sample time — шаг модельного времени (интервал дискретизации при отображении сигнала); Scroll — непрерывное прокручивание осциллограммы во время моделирования; Legends — отображение легенды; Floating Scope — изменяющийся режим (перевод осциллографа в изменяющийся режим).

На закладке History задаются следующие параметры: Limit data points to last — максимальное количество отображаемых расчетных точек графика (если этот флажок не установлен, то количество отображаемых точек определяется количеством расчетных значений); Save data to workspace — сохранить расчетные значения сигналов в рабочем пространстве MATLAB; Variable name — имя переменной для сохранения сигналов в рабочем пространстве; Format — формат данных для сохранения сигналов в рабочем пространстве: Array — массив; Structure — структура (массив записей); Structure with time — структура (массив записей) с дополнительным полем времени.

В тестовом режиме TimeScope есть возможность управлять моделированием, использовать маркеры, триггеры, измерять статистические параметры и использовать больше графических настроек.





### **Задание к лабораторной работе**

1. Создать новую модель, которая должна вычислять одно из алгебраических выражений Лабораторной работы 1.
2. С помощью блоков библиотеки Simulink/Math Operations собрать модель, вычисляющую соответствующее выражение.
3. Значения некоторых переменных ввести в блоках Constant, а остальных — в скрипте MATLAB.
4. Вычисленное выражение подключить к блоку Display. Рядом на блок Display с помощью блока From Workspace вывести соответствующее значение  $y$  или  $z$ , полученное на первом практическом занятии. Сравнить результаты.

### **Второе задание к лабораторной работе**

1. Создать новую модель и добавить блок Subsystem, у которого входные порты  $F_s$ ,  $V_{in}$ ,  $i_0$  и выходные порты  $V_c$  и  $i_L$ .
2. В подсистеме собрать модель, решающую систему дифференциальных уравнений:

$$\begin{cases} C \frac{dV_c}{dt} = i_L - \frac{V_c}{R} - i_0, \\ L \frac{di_L}{dt} = F_s V_{in} - V_c \end{cases}$$

3. Значения переменных  $R$ ,  $C$  и  $L$  ввести в блоках Constant внутри Subsystem в соответствии с вариантом. Выходные порты  $V_c$  и  $i_L$  подключить к блоку Scope. Входные порты  $V_{in}$  и  $i_0$  подключить к блокам Constant.  $V_{in} = 3.3$  В. Порт  $F_s$  подключить к блоку Pulse Generator. Период импульсов 1 мкс, амплитуда 1.
4. Подобрать ширину импульсов так, чтобы  $V_c = V_{out}$ .

Вариант	Задание
1	$R = 10$ Ом, $C = 5$ мкФ, $L = 5$ мкГн, $i_0 = 200$ мА, $V_{out} = 1.1$ В
2	$R = 15$ Ом, $C = 2$ мкФ, $L = 7$ мкГн, $i_0 = 100$ мА, $V_{out} = 1.2$ В
3	$R = 12$ Ом, $C = 3$ мкФ, $L = 10$ мкГн, $i_0 = 300$ мА, $V_{out} = 1.3$ В
4	$R = 18$ Ом, $C = 6$ мкФ, $L = 12$ мкГн, $i_0 = 200$ мА, $V_{out} = 1.4$ В
5	$R = 12$ Ом, $C = 5$ мкФ, $L = 9$ мкГн, $i_0 = 100$ мА, $V_{out} = 1.5$ В

6	$R = 15 \text{ Ом}, C = 3 \text{ мкФ}, L = 6 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 1.6 \text{ В}$
7	$R = 10 \text{ Ом}, C = 6 \text{ мкФ}, L = 6 \text{ мкГн}, i_0 = 300 \text{ мА}, V_{\text{out}} = 1.7 \text{ В}$
8	$R = 15 \text{ Ом}, C = 7 \text{ мкФ}, L = 8 \text{ мкГн}, i_0 = 100 \text{ мА}, V_{\text{out}} = 1.8 \text{ В}$
9	$R = 12 \text{ Ом}, C = 8 \text{ мкФ}, L = 11 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 1.9 \text{ В}$
10	$R = 18 \text{ Ом}, C = 10 \text{ мкФ}, L = 8 \text{ мкГн}, i_0 = 300 \text{ мА}, V_{\text{out}} = 2.0 \text{ В}$
11	$R = 10 \text{ Ом}, C = 8 \text{ мкФ}, L = 15 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 2.1 \text{ В}$
12	$R = 15 \text{ Ом}, C = 8 \text{ мкФ}, L = 12 \text{ мкГн}, i_0 = 100 \text{ мА}, V_{\text{out}} = 2.2 \text{ В}$
13	$R = 18 \text{ Ом}, C = 3 \text{ мкФ}, L = 7 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 1.1 \text{ В}$
14	$R = 12 \text{ Ом}, C = 9 \text{ мкФ}, L = 12 \text{ мкГн}, i_0 = 300 \text{ мА}, V_{\text{out}} = 1.2 \text{ В}$
15	$R = 10 \text{ Ом}, C = 7 \text{ мкФ}, L = 10 \text{ мкГн}, i_0 = 100 \text{ мА}, V_{\text{out}} = 1.3 \text{ В}$
16	$R = 15 \text{ Ом}, C = 4 \text{ мкФ}, L = 12 \text{ мкГн}, i_0 = 300 \text{ мА}, V_{\text{out}} = 1.4 \text{ В}$
17	$R = 12 \text{ Ом}, C = 6 \text{ мкФ}, L = 11 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 1.5 \text{ В}$
18	$R = 18 \text{ Ом}, C = 8 \text{ мкФ}, L = 12 \text{ мкГн}, i_0 = 100 \text{ мА}, V_{\text{out}} = 1.6 \text{ В}$
19	$R = 9 \text{ Ом}, C = 10 \text{ мкФ}, L = 10 \text{ мкГн}, i_0 = 200 \text{ мА}, V_{\text{out}} = 1.7 \text{ В}$
20	$R = 15 \text{ Ом}, C = 6 \text{ мкФ}, L = 15 \text{ мкГн}, i_0 = 100 \text{ мА}, V_{\text{out}} = 1.8 \text{ В}$

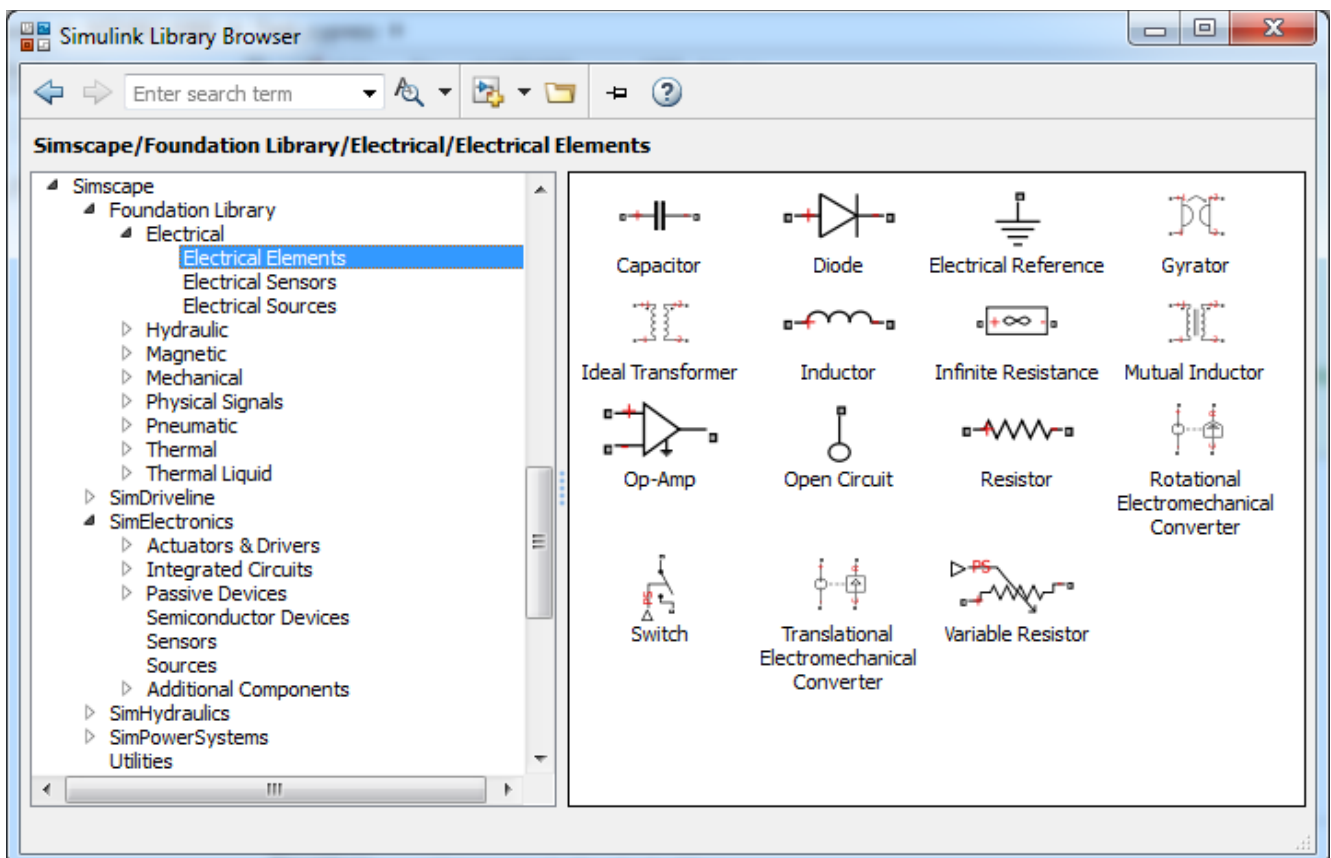
## **Лабораторная работа 13. Моделирование электронных схем в SIMULINK.**

### **Основные элементы библиотеки Simscape**

Библиотека Simscape предоставляет собой набор фундаментальных блоков для построения и симуляции физических систем, содержащих компоненты из различных инженерных сфер деятельности: механических, электрических, гидравлических и других.

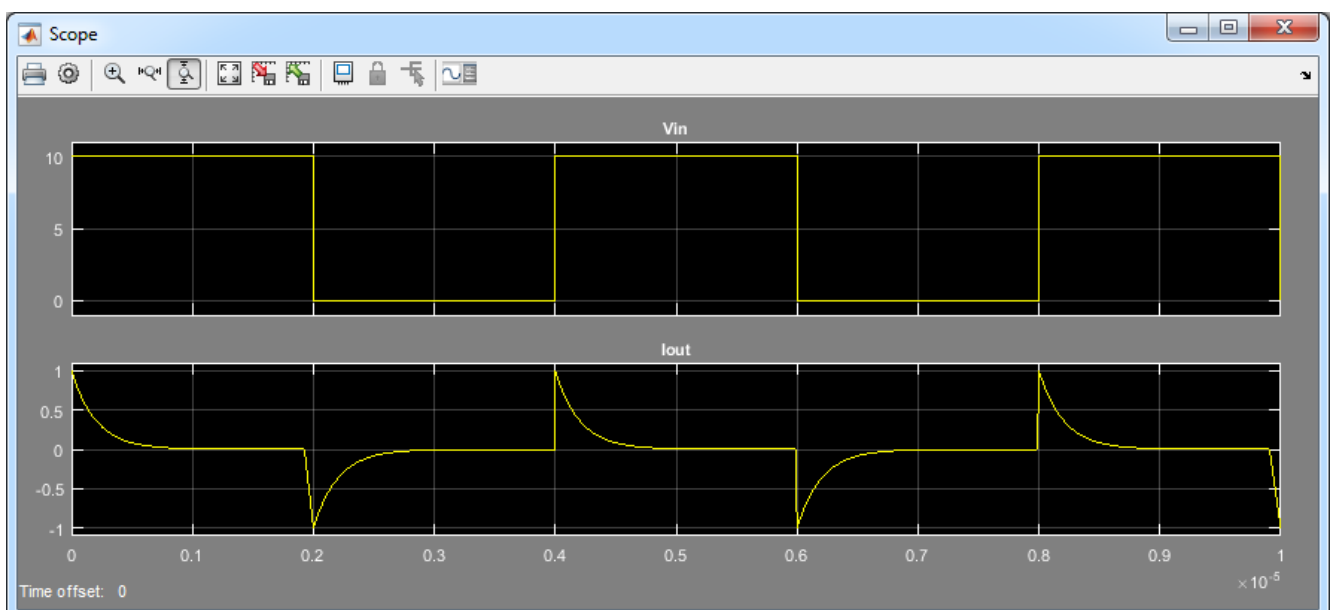
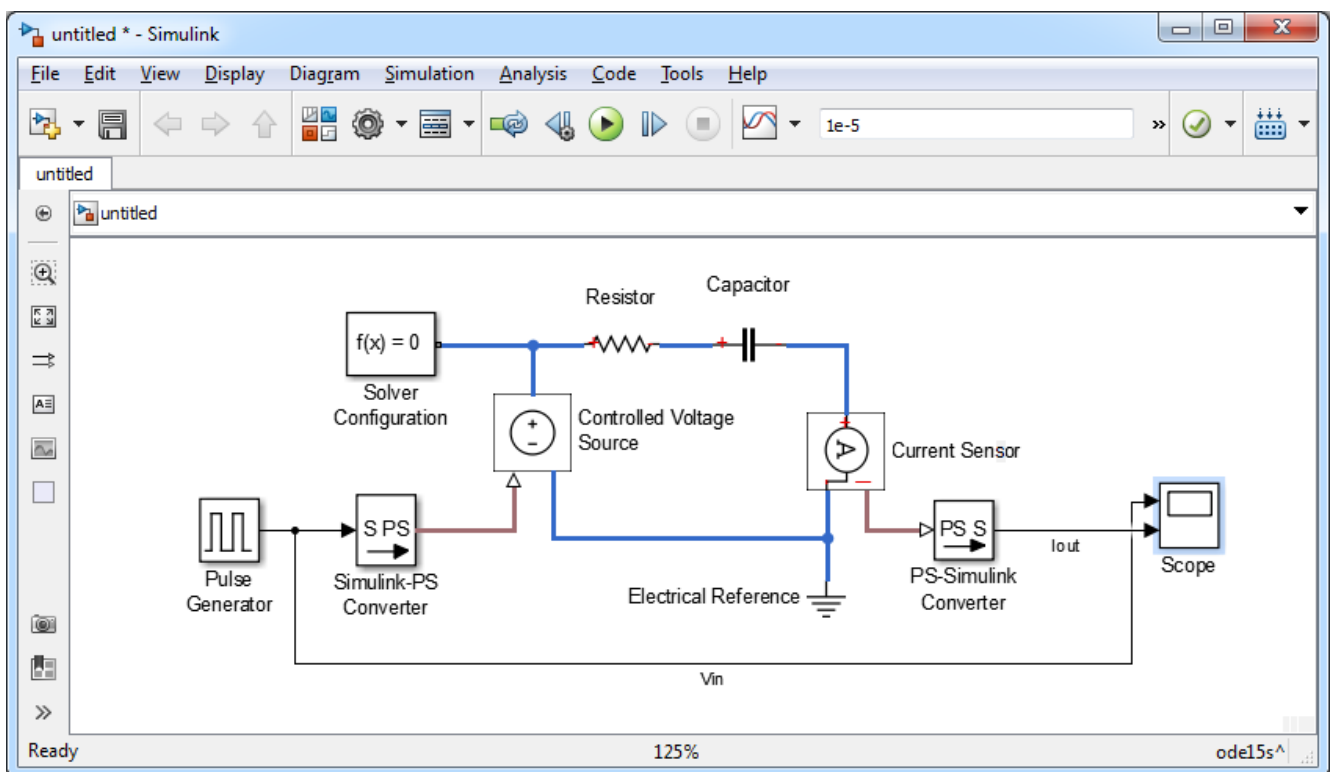
Для построения модели Simscape использует подход, называемый "физическая сеть", также известный как каузальное моделирование: компоненты (блоки), относящиеся к физическим элементам, таким, как насосы, двигатели и операционные усилители, соединяются линиями, представляющими физические соединения, по которым передается энергия. Этот подход позволяет описывать физическую структуру системы.

Simscape автоматически выводит дифференциальные уравнения, характеризующие поведение системы. Эти уравнения совмещаются с остальной моделью SIMULINK и решаются напрямую. Компоненты из различных физических доменов решаются совместно, относительно своих переменных, таким образом, позволяя избегать алгебраических зацикливаний.



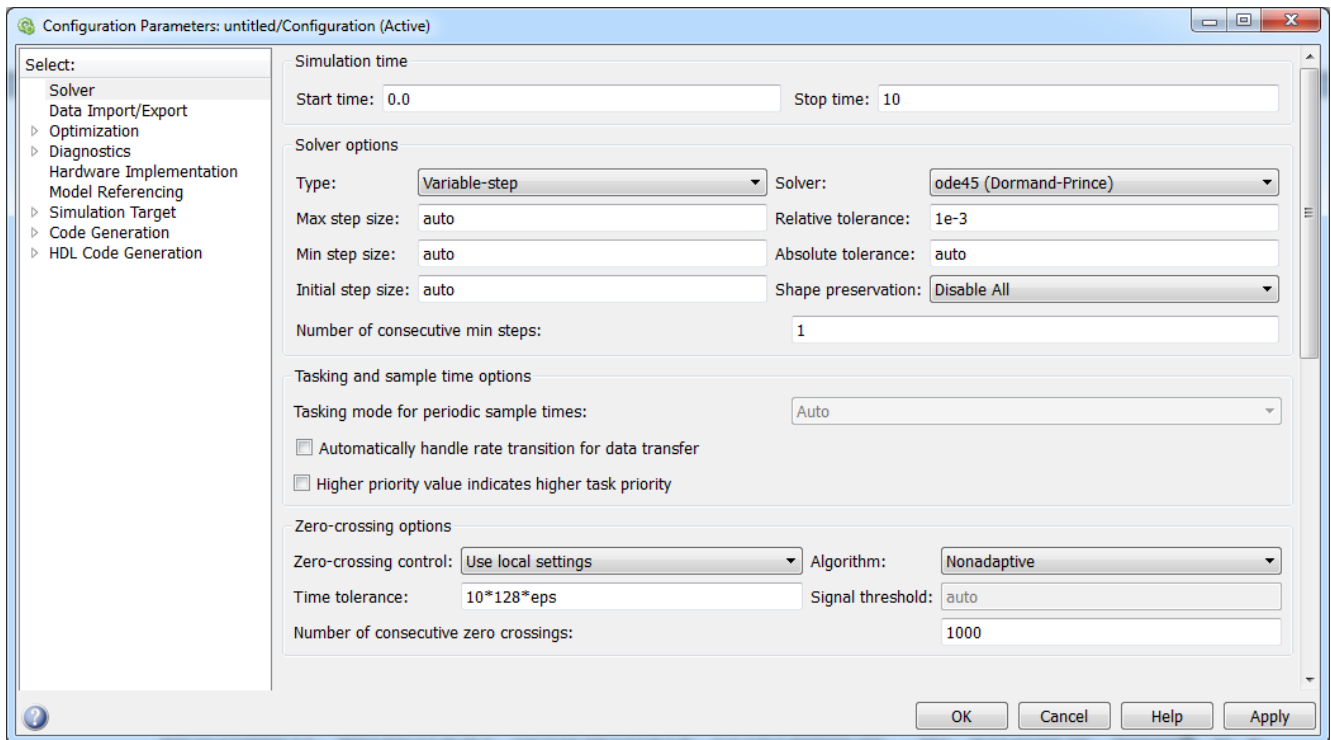
Для моделирования электронных схем используются блоки из разделов Simscape/Foundation Library/Electrical и Simscape/SimElectronics. На модели электронной схемы обязательно должны быть блоки Electrical Reference (Foundation Library/Electrical/Electrical Elements) и Solver Configuration (Utilities). Причем, последний подключается к любому месту в схеме.

Чтобы преобразовать сигналы из SIMULINK в физические сигналы Simscape используются блоки PS-Simulink Converter и Simulink-PS Converter из раздела Utilities. Далее приведен пример модели с использованием блоков Simscape:



## Установка параметров моделирования

Перед моделированием при необходимости можно задать основные параметры анализируемых процессов. Для этого в окне модели выполняются действия в меню “Simulation” → “Model Configuration Parameters”:



Появляется окно, имеющее закладки: Solver — решатель для установки параметров моделирования; Data Import/Export — ввод-вывод данных в рабочую область; Diagnostics — настройка параметров диагностирования и закладки для генерации кода и подключения дополнительного оборудования.

В закладке решателя Solver имеются группы параметров:

Simulation time — интервал моделирования указывается в виде начального (Start time — обычно нулевое значение) и конечного (Stop time) значений времени;

Solver options — параметры решателя, определяемые методом интегрирования (Type) с фиксированным (Fixed-step) или с переменным (Variable-step) шагом. Важный момент при установке параметров решателя — это выбор конкретного метода интегрирования (Solver) при решении дифференциальных уравнений: ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb, которые отличаются друг от друга скоростью расчета и погрешностью получаемого решения. После списков Type и Solver приводится область, содержание которой

зависит от выбора Fixed-step или Variable-step. При выборе Variable-step появляются поля для установки следующих параметров:

Max step size — максимальный шаг интегрирования. По умолчанию этот параметр выбирается автоматически (auto) и составляет 0,02 от величины времени моделирования, то есть  $(\text{Stop time} - \text{Start time})/50$ . Иногда этот шаг оказывается больше, чем это требуется, и тогда построенные временные диаграммы оказываются состоящими из ломаных линий и в значительной мере отличаются от ожидаемых кривых. По выбору шага интегрирования далее будут даны соответствующие рекомендации.

Min step size — минимальный шаг интегрирования.

Initial step size — начальная величина шага интегрирования.

Погрешности вычислений при моделировании непрерывных систем по умолчанию задаются равными — относительная (Relative tolerance)  $1e-3$  и абсолютная (Absolute tolerance) auto или устанавливаются в требуемых пределах.

В закладке Data Import/Export осуществляется управление вводом в рабочее пространство и выводом из него результатов моделирования. Имеются поля:

Load from workspace — загрузить из рабочей области. При установленном флажке Input (Входные данные) в текстовое поле вводится формат данных, которые будут считываться из рабочего пространства. Флажок Initial State (Начальное состояние) разрешает ввод в текстовое окно имени переменной. Все указанные данные передаются в модель с помощью блока(ов) In, находящихся в разделе Sources.

Save to workspace — запись данных, состояний в рабочее пространство.

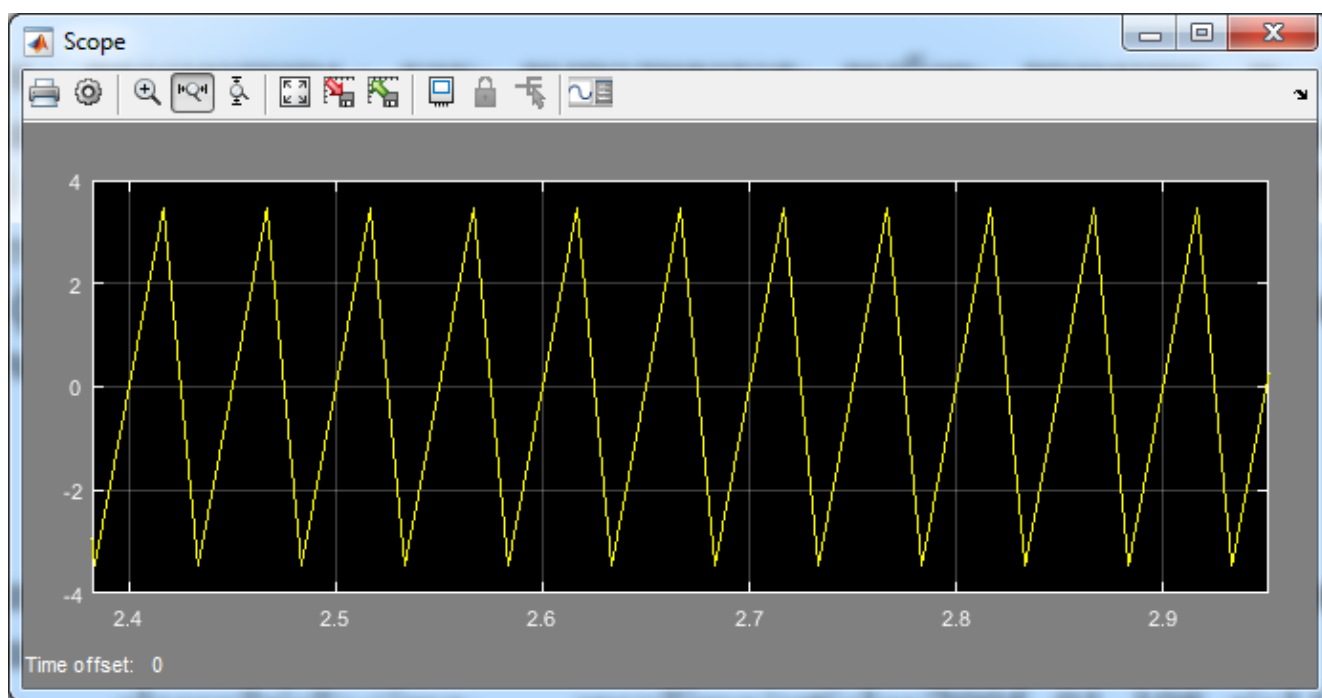
В нижней части настраиваются параметры вывода выходного сигнала моделируемой системы (Save options):

Refine output — скорректированный вывод для изменения шага регистрации модельного времени и тех сигналов, которые сохраняются в рабочем пространстве при использовании блока To Workspace. Установка периода дискретизации осуществляется в строке Refine factor. Установка по умолчанию — 1 (регистрируется каждый шаг) или задается положительное целое число  $n$  (регистрируется каждое  $n$ -е значение);

Produce additional output — дополнительный вывод, обеспечивающий дополнительную регистрацию параметров модели в заранее заданные моменты времени, которые вводятся в строку редактирования (Output times) в виде списка в квадратных скобках. Значения времени могут быть дробными.

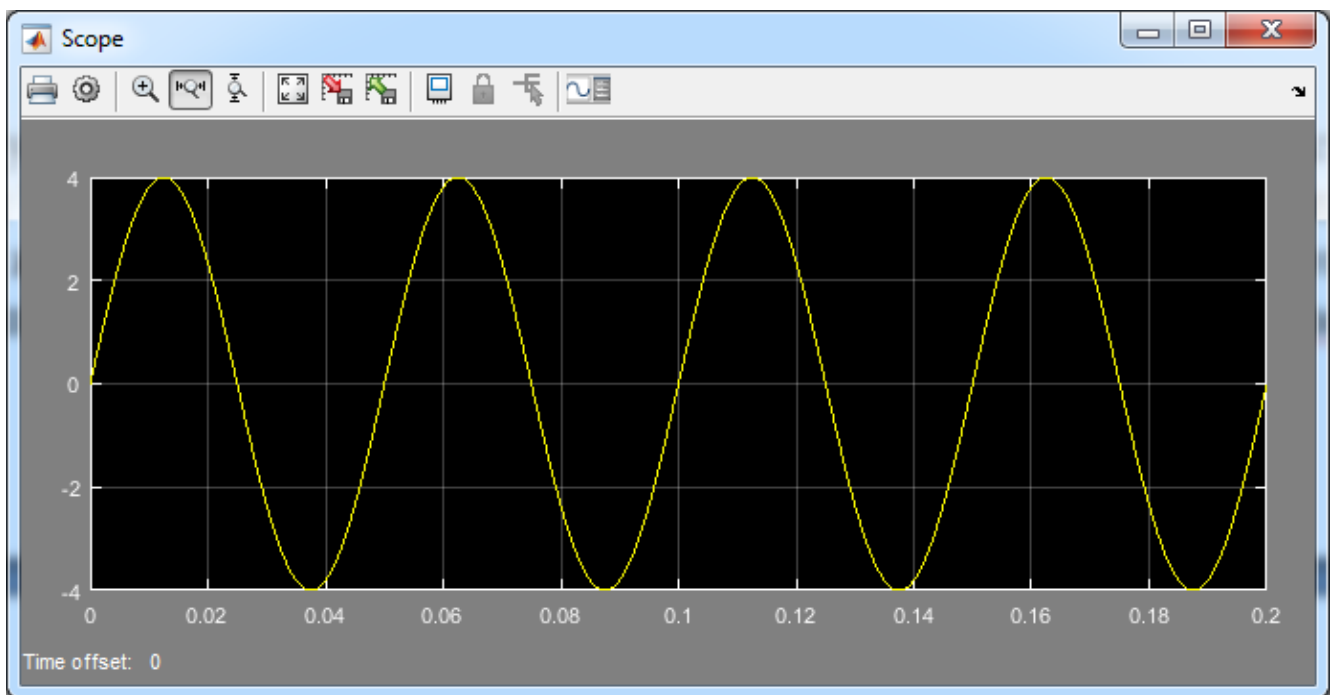
Produce specified — редактирование заданного вывода для установки вывода параметров модели в заданные в виде вектора в поле Output times моменты времени.

Теперь рассмотрим, как выполняется выбор времени и шага моделирования. Предположим, нужно получить синусоидальный сигнал с амплитудой 4 В и частотой 20 Гц. Собираем схему из генератора Sin Wave (настраиваем его амплитуду и частоту, период дискретизации равен 0) и осциллографа Scope и запускаем полученную модель. В окне осциллограммы Scope получается временная диаграмма, далекая от синусоиды:



Причина этого явления объясняется неправильно выбранными параметрами моделирования.

Для установки параметров моделирования необходимо знать частоту или период получаемого сигнала. Если этот сигнал сложный, то следует иметь те же параметры для его высокочастотной и низкочастотной составляющих. В окне настроек модели Model Configuration Parameters вызывается закладка Solver. Шаг моделирования выбирается из простого требования — минимальный период сигнала должен на порядок или более превышать максимальный шаг моделирования (шаг дискретизации). В нашем случае при  $T = 1/20 = 0,05$  шаг выбирается 0,002 или меньше. Указанное значение шага набирается в текстовом окне Max step size. Время моделирования считаем из условия — в осциллограмме должно быть, например, 4 больших периода, то есть в текстовом окне Stop time устанавливается 0,2. После запоминания установленных параметров и запуска моделирования получаем приемлемую осциллограмму:

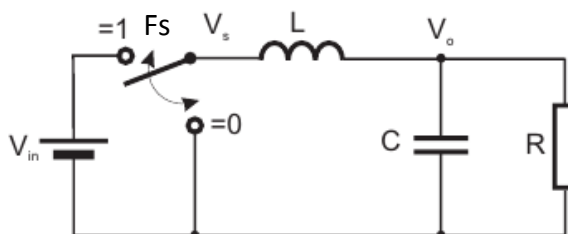


Итак, для расчета процессов в функциональной модели необходимо произвести настройки с учетом минимальной и максимальной частот обрабатываемого сигнала, выбрать максимальный шаг дискретизации и время моделирования, а затем контролировать получаемые сигналы по временной диаграмме.



### Задание к лабораторной работе

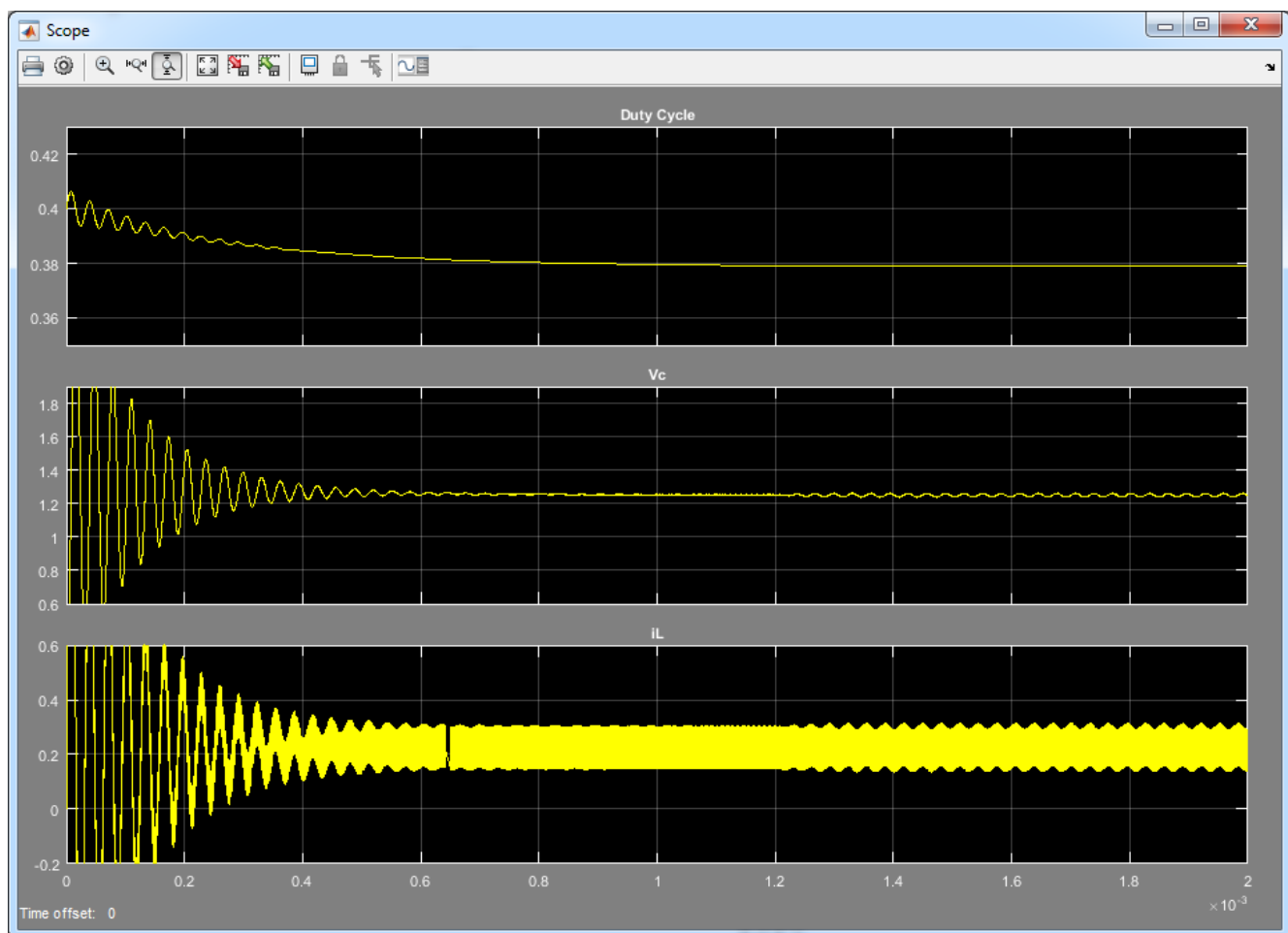
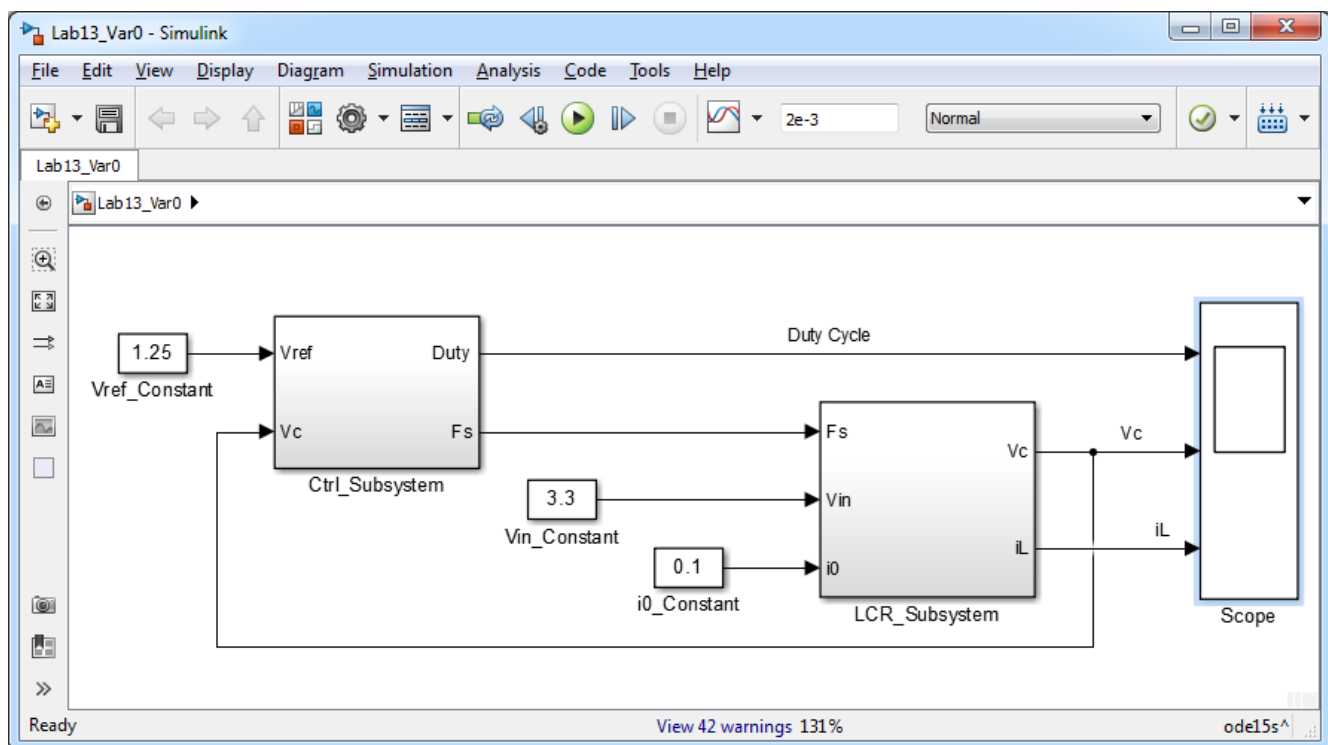
1. Создать новую модель и добавить блок LCR\_Subsystem, у которого входные порты  $F_s$ ,  $V_{in}$ ,  $i_0$  и выходные порты  $V_c$  и  $i_L$ .
2. В этой подсистеме собрать модель на элементах Simscape, реализующую следующую электрическую схему:



3. Значения переменных  $R$ ,  $C$  и  $L$  взять из Лабораторной работы 12 в соответствии с вариантом. Выходные порты  $V_c$  и  $i_L$  подключить к блоку Scope. Входные порты  $V_{in}$  и  $i_0$  подключить к блокам Constant.  $V_{in} = 3.3$  В. Порт  $F_s$  подключить к блоку Pulse Generator. Период импульсов 1 мкс, амплитуда 1.
4. В настройках Configuration Parameters установить Solver = ode15s.
5. Подобрать ширину импульсов Pulse Generator так, чтобы  $V_c = V_{out}$ .
6. Добавить блок Ctrl\_Subsystem, который будет формировать сигнал  $F_s$  вместо блока Pulse Generator. Его выходной порт  $F_s$  подключить к соответствующему порту LCR\_Subsystem.
7. В блок Ctrl\_Subsystem добавить Counter Limited, Relational Operator и другие необходимые элементы. Собрать схему так, чтобы период сигнала  $F_s$  был 1 мкс, а ширина импульсов задавалась постоянной величиной от 0 до 1.
8. Добавить входные порты  $V_c$  и  $V_{ref}$ . Порт  $V_c$  подключить к соответствующему порту LCR\_Subsystem. На порт  $V_{ref}$  подать значение напряжения  $V_{out}$ , которое нужно получить на выходе схемы.
9. Разность сигналов  $V_{ref}$  и  $V_c$ , увеличенную в 1000 раз, подать на Integrator. Полученный сигнал использовать в качестве значения ширины импульсов для  $F_s$ .
10. Запустить моделирование при разных  $V_{ref}$  и посмотреть, как схема автоматически подстраивает значение  $V_c$ .

Для коммутации входных сигналов на этапе отладки модели удобно использовать блок Manual Switch из раздела Simulink/Signal Routing.

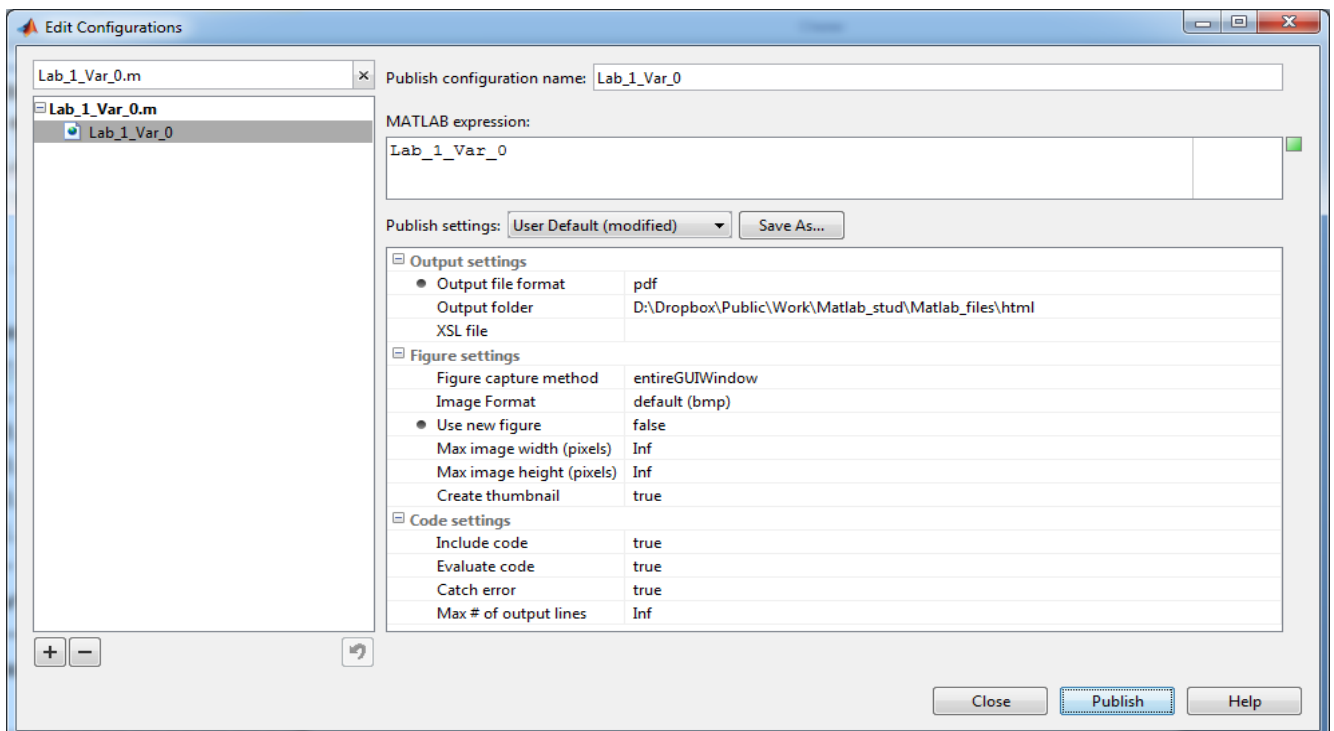
Пример выполнения лабораторной работы:



## Пример m-файла для создания отчета

```
%% Lab_X Var_X
% First_name Second_name, Group_number
%
%% Task_1
clear; clc;
%---- INPUT DATA ----
a = 3.1;
k = 0.5;
b = 0.15;
x = 1.25;
%---- OUTPUT DATA ----
y = a*x^2/cos(b) + sqrt(exp(x)) - 5*k
%-----
%% Task_2
...
%-----
```

Редактируем настройки PUBLISH → Publish → Edit Publishing Options... (в MATLAB R2009 File → Publish Configuration → Edit Publish Configurations) и нажимаем кнопку Publish:



Или создать и запустить m-файл следующего содержания:

```
options_pdf.format = 'pdf';
options_pdf.useNewFigure = false;
publish('Lab1_var0',options_pdf)
```