

Министерство образования и науки Российской Федерации  
Санкт–Петербургский политехнический университет Петра Великого  
Институт энергетики  
Высшая школа высоковольтной энергетики

Кафедра электрических станций и автоматизации энергетических систем

К.С. Горячевский, М. Г. Попов  
КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ. ПРОГРАММА МАТЛАВ.  
Методическое пособие для практических занятий

Санкт–Петербург  
2021

## Оглавление

1. Установка MATLAB .....	3
2. Знакомство с программой .....	8
3. MATLAB как калькулятор .....	9
4. Логические операции.....	12
5. Встроенные и пользовательские функции .....	13
6. Матрицы, вектора и массивы.....	17
7. Циклы и условия .....	20
8. Вывод графиков.....	23
9. Расчётное задание .....	25
9.1. Решение квадратного уравнения.....	25
9.2. Вычислить логическое выражение.....	25
9.3. Работа с массивом.....	26
9.4. Работа с комплексными числами .....	27
10. Варианты заданий .....	28

## 1. Установка MATLAB

Студенты СПбПУ могут бесплатно использовать программу MATLAB, если зарегистрируются с помощью почты @edu.spbstu.ru.

Для установки программы запустите скачанный файл «matlab.exe». После загрузки программы появится окно на рисунке 1.1, где нужно нажать кнопку «Create one» и создать учётную запись MATLAB.

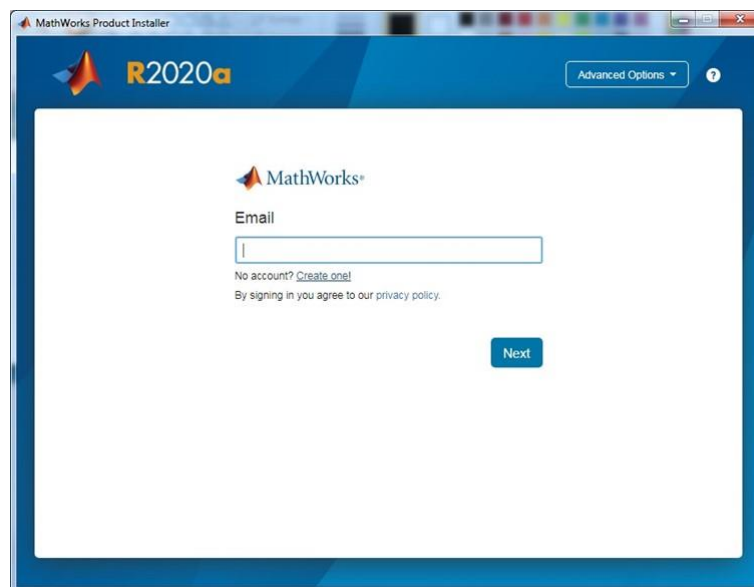


Рисунок 1.1 — Начало установки

На рисунке 1.2 показан пример заполнения учётных данных пользователя. Введите свои имя, фамилию, почту СПбПУ и нажмите кнопку «Continue».

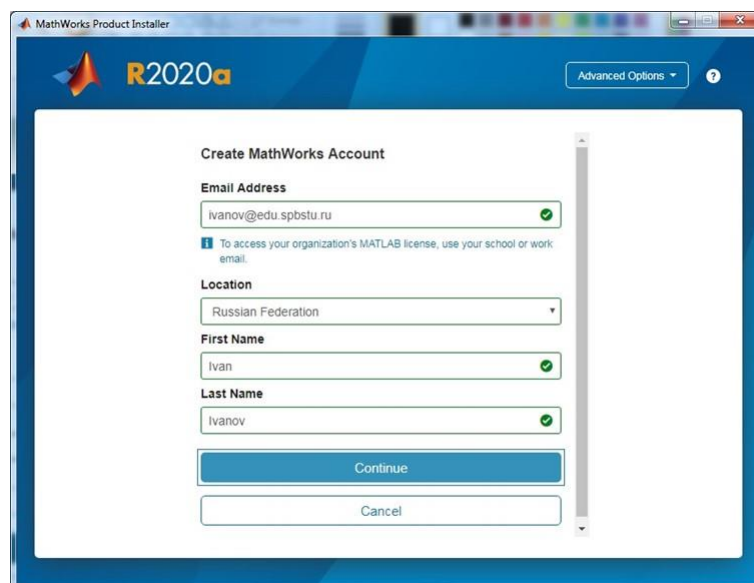


Рисунок 1.2 — Учетные данные

После этого необходимо проверить электронную почту и открыть ссылку в письме с подтверждением регистрации. Затем вернуться к программе установки и на экране на рисунке 1.3 нажать кнопку «Continue» и придумать пароль.

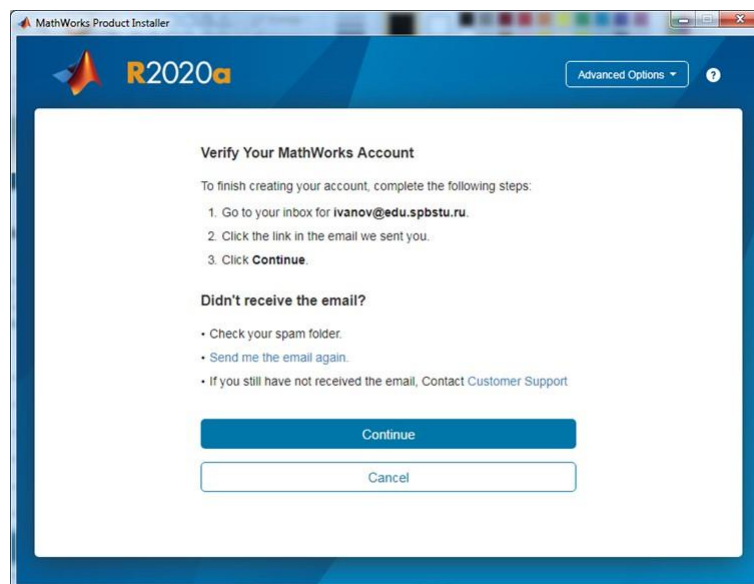


Рисунок 1.3 — Проверка электронной почты

После регистрации необходимо ввести логин, как это показано на рисунке 1.4, нажать кнопку «Next» и ввести пароль.

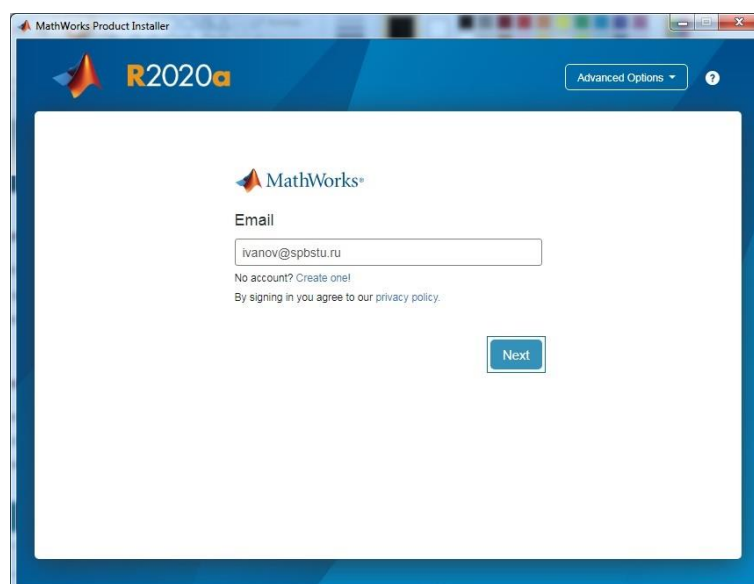


Рисунок 1.4 — Вход с учётной записью

На следующем экране, который показан на рисунке 1.5, необходимо согласиться с лицензией на программу, нажав кнопку «Yes» и «Next».

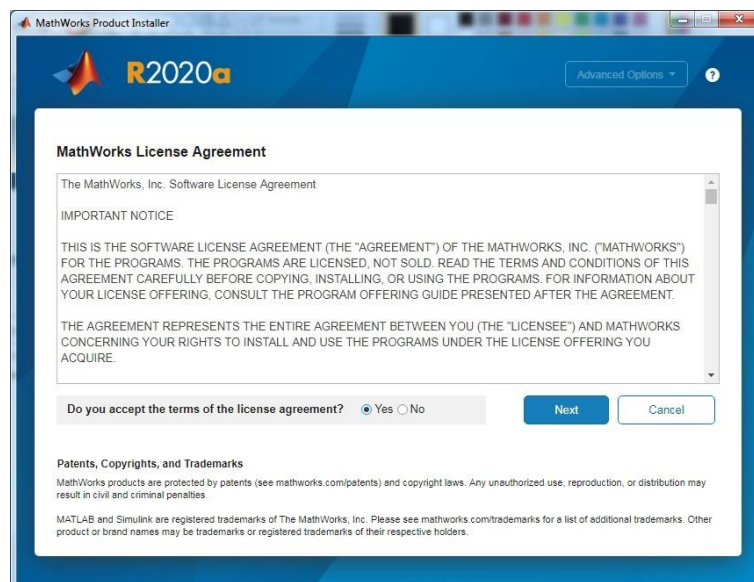


Рисунок 1.5 — Лицензия на программу

На следующих экранах, показанных на рисунках 1.6, 1.7, 1.8, необходимо нажать «Next».

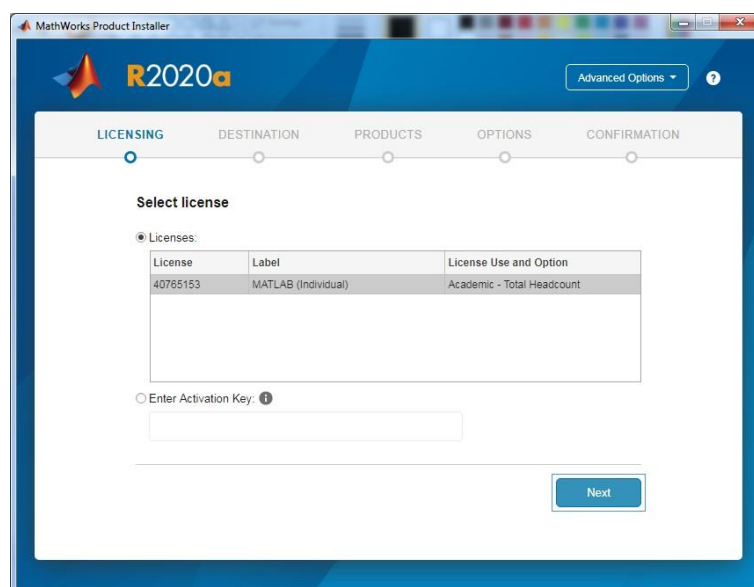


Рисунок 1.6 — Ключ лицензии к программе

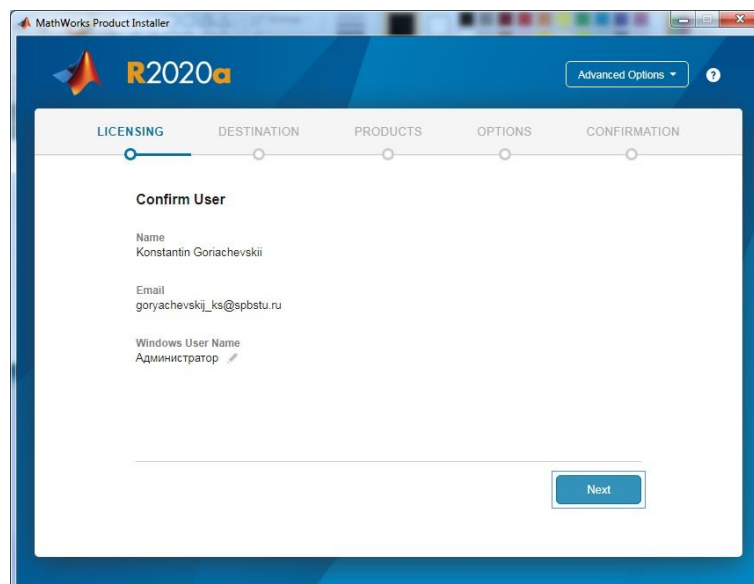


Рисунок 1.7 — Проверка учётных данных

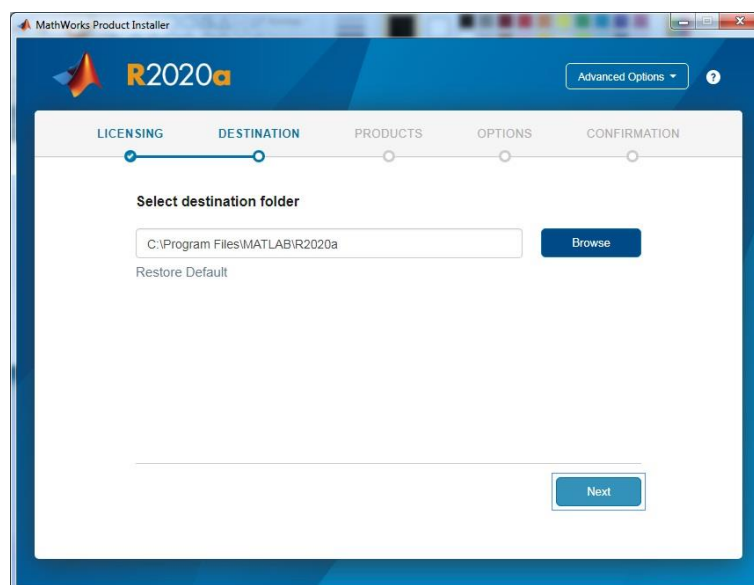


Рисунок 1.8 — Файлы программы на диске

На экране, показанном на рисунке 1.9, можно выбрать функции MATLAB, которые будут установлены. Для лабораторных работ потребуется только первая функция, это сама программа MATLAB, все остальные галочки можно снять.

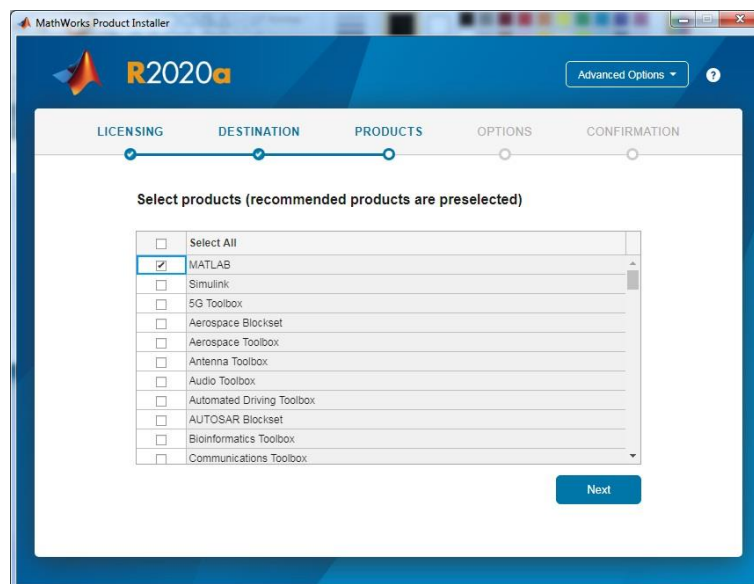


Рисунок 1.9 — Выбор необходимых функций

На экране на рисунке 1.10 необходимо поставить первую галочку «Add shortcut to desktop», чтобы ярлык программы появился на рабочем столе, и нажать кнопку «Next».

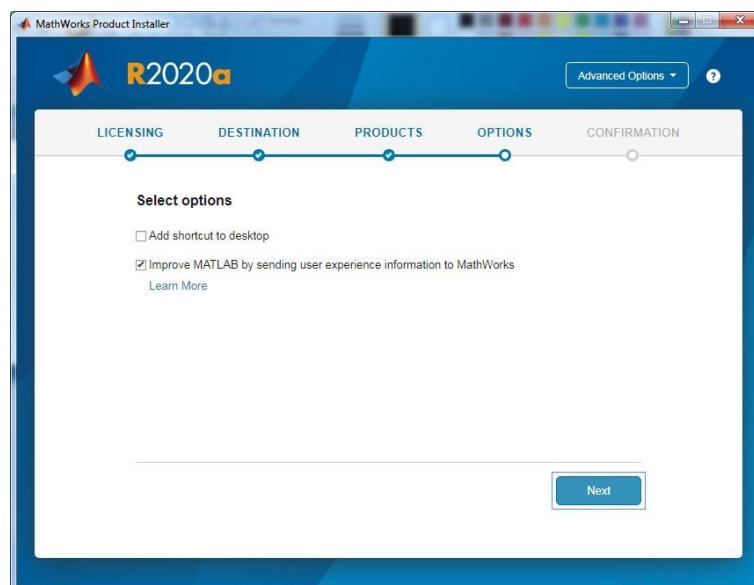


Рисунок 1.10 — Дополнительные настройки

После нажатия кнопки «Begin install» на экране на рисунке 1.11 начнётся установка программы. В зависимости от компьютера это может занять до одного часа.

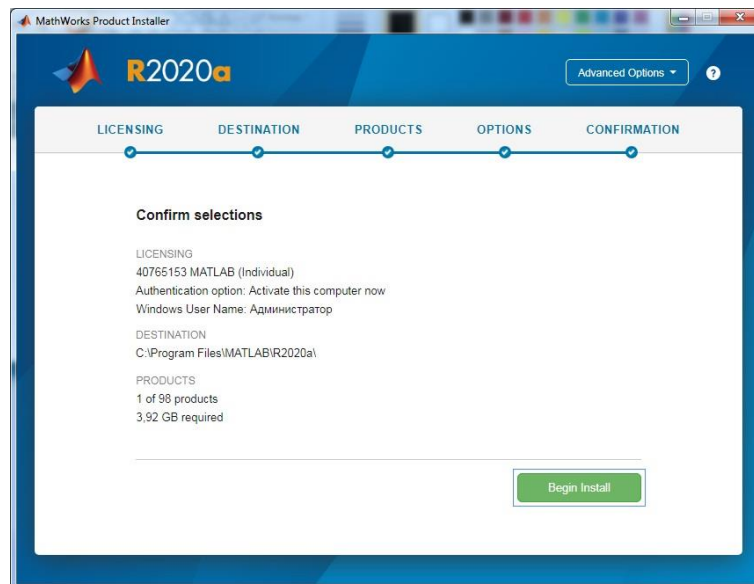


Рисунок 1.11 — Начало установки

## 2. Знакомство с программой

На рисунке 2.1 показан интерфейс программы. Программа работает только на английском языке, но все действия в программе типовые, поэтому навык работы вырабатывается сравнительно легко.

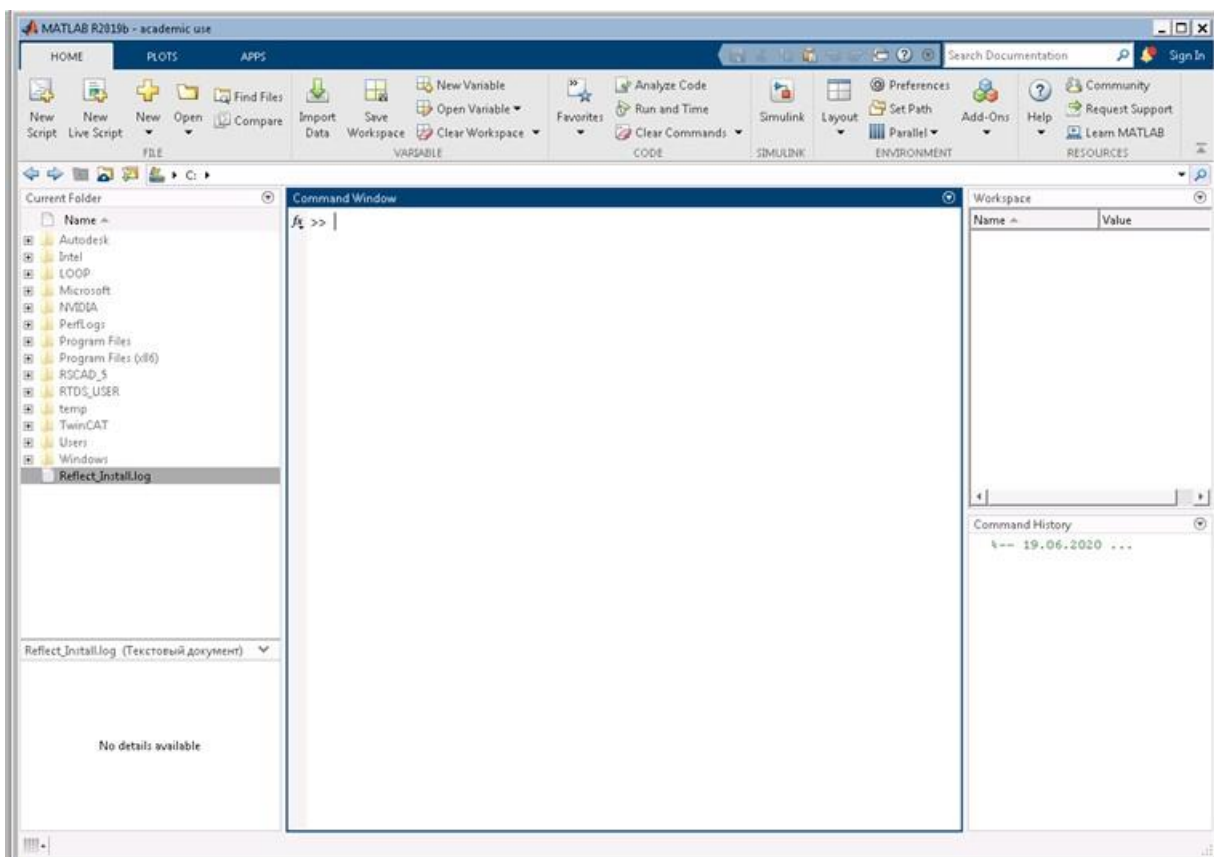


Рисунок 2.1 — Программа MATLAB

На экране показаны следующие окна и кнопки:



- в центре находится окно «Command Window», в которое вводятся команды с клавиатуры, это окно является самым главным в программе;
- слева находится окно «Current Folder», в котором можно найти и открыть файлы MATLAB;
- справа находится окно «Workspace», в котором отображаются переменные и их значения;
- также справа находится окно «Command History», в котором появляются команды с клавиатуры, введённые ранее;
- сверху находятся кнопки «Новый файл», «Открыть файл», «Настройки программы» (Preferences, значок «шестерёнка») и т.п.

Если шрифт на экране маленький, то нажмите кнопку «Preferences → Fonts → введите размер шрифта → ОК».

Чтобы ввести свою первую команду, нажмите ЛКМ (левой кнопкой мыши) в центре экрана, чтобы выбрать окно команд (надпись «Command Window» станет синей) и введите команду

```
>> 2+2
```

после чего нажмите на клавиатуре «Ввод» (символ >> вводить не надо). На экране появится результат

```
ans =  
      4
```

Поздравляем, вы только что выполнили свою первую программу на языке MATLAB!

Обратите внимание, что в правой части экрана появилась переменная *ans* = 4 и команда 2 + 2. Чтобы повторить команду, нажмите на клавиатуре кнопку «Вверх».

Чтобы очистить экран, введите команду

```
>> clc
```

Чтобы очистить окно «Workspace», в котором хранятся переменные, введите команду

```
>> clear
```

Если какие-то символы или команды введены неправильно, то MATLAB покажет красный текст в окне команд, где подробно описана ошибка.

### 3. MATLAB как калькулятор

Выполните следующие команды для того, чтобы познакомиться с простейшими арифметическими операциями в MATLAB.

Задание новой переменной

```
>> A = 1
```

Переменные должны иметь в названии только латинские буквы и цифры, начинаться с буквы. При записи десятичных дробей используют точку.

На экране появится результат

A =

1

Обратите внимание, что справа появилась новая переменная  $A = 1$ .

Введите следующую команду, но добавьте в конец точку с запятой

>> B = 2;

При этом результат не отобразится в окне команд, но появится справа в окне переменных.

Сложите эти значения:

>> C = A + B

Вычитание:

>> D = A - B

Умножение:

>> E = A \* B

Деление:

>> F = A / B

Возведение в степень:

>> G = C ^ 2

Извлечение квадратного корня (название команды sqrt от «square root»)

>> H = sqrt(C)

или через показатель степени

>> H = C^(1/2)

При записи длинных выражений можно использовать скобки для того, чтобы указать порядок выполнения операций. Если скобки не расставлены, то выражение выполняется следующим образом:

- сначала возведение в степень и извлечение корня;
- затем умножение и деление;
- сложение и вычитание;
- если в выражении несколько одинаковых операций, они выполняются слева направо.

Введите сложное выражение:

>> X = ( B ^ 2 + C ^ -3 / 2 - 1 ) / ( D \* F )

Это соответствует формуле

$$X = \frac{B^2 + \frac{C^{-3}}{2} - 1}{D \cdot F}$$

Результат выполнения

```
X =  
-6.0370
```

При записи дроби обычно числитель и знаменатель выделяют скобками

```
>> Y = ( A + B ) / ( C + D )
```

Результат выполнения

```
Y =  
1.5000
```

Также MATLAB может использоваться для работы с комплексными числами. Мнимая часть комплексного числа записывается как число и символ  $i$  без пробела. Пример:

```
>> Z = ( 1 + 1i ) / ( 0.1 - 0.2i )
```

Результат выполнения

```
Z =  
-2.0000 + 6.0000i
```

Операция натурального логарифма

```
>> log ( 2.7183 ) ;
```

Операция десятичного логарифма

```
>> log10 ( 10 ) ;
```

Логарифм  $b$  по основанию  $a$ :

```
>> log ( b ) / log ( a ) ;
```

Экспонента:

```
>> exp ( 1 ) ;
```

Модуль:

```
>> abs ( -1 ) ;
```

Число Пи:

```
>> pi
```

Тригонометрические функции:

```
>> cos ( pi )
```

```
>> sin ( pi )
```

```
>> tan ( pi )
```

```
>> cot ( pi )
```

```
>> acos ( 1 )
```

```
>> asin ( 1 )
>> atan ( 1 )
>> acot ( 1 )
```

Результат *Inf* обозначает *infinity*, т.е. бесконечность.

В программе широко используется инженерная форма записи чисел, когда после числа пишется буква *e* и целое число, что обозначает «умножить на десять в этой степени», например:

$$1.1e-1 = 1.1 \cdot 10^{-1} = 0.11$$
$$2.2e2 = 2.2 \cdot 10^2 = 220$$

Числа меньше  $1e-3$  очень близки к нулю и часто могут быть округлены до нуля. Числа больше  $1e10$  часто могут пониматься как бесконечность и могут обозначать неправильный расчёт, например, деление на число, близкое к нулю.

#### 4. Логические операции

Логическими операциями называются правила работы с числами 1 и 0. Также математическая логика называется алгеброй Буля (булева алгебра) в честь английского математика Джорджа Буля (1815-1864).

Для компьютеров булева алгебра является очень важной, т.к. компьютеры работают только с двоичными числами. Основные операции булевой алгебры: И, ИЛИ, НЕ. Также встречаются ИСКЛЮЧАЮЩЕЕ ИЛИ, ИЛИ-НЕ, И-НЕ, но они состоят из основных операций.

Числа 1 и 0 также обозначаются как ИСТИНА и ЛОЖЬ или TRUE и FALSE.

Операция & (также И, AND, &&, ·) является простым умножением, результат ИСТИНА только когда обе переменные ИСТИНА. Выполните в MATLAB следующие операции и обратите внимание на результат:

```
>> 0 & 0
>> 1 & 0
>> 0 & 1
>> 1 & 1
```

Операция | (вертикальная черта, также ИЛИ, OR, ||, на графических схемах цифра 1) является простым сложением, результат ИСТИНА когда любая из переменных ИСТИНА. Выполните снова эти операции и посмотрите результат.

Операция ~ (тильда, также НЕ, NOT, !, горизонтальная черта над символом, на графических схемах кружок) является инверсией и заменяет ИСТИНА на ЛОЖЬ и наоборот. Эта операция производится только над одной переменной (унарная операция). Выполните следующие команды:

```
>> ~1
```

```
>> ~0
```

Операция `==` (два знака "равно") возвращает результат ИСТИНА, когда переменные слева и справа равны, и ЛОЖЬ, когда переменные не равны.

Операция `xor` (также ИСКЛЮЧАЮЩЕЕ ИЛИ,  $\oplus$ , на графических схемах  $=1$ ) является сложением с переносом, результат ИСТИНА только когда одна переменная ИСТИНА, а другая ЛОЖЬ. Выполните в MATLAB операции, которые выполняли для И.

Часто в программировании в логических операциях участвуют переменные, значение которых не равно 0 или 1, а может быть любым другим числом. Все эти числа приравниваются к 1. Выполните команды:

```
>> 2 & 0
>> 3 | 0.1
>> ~-5
>> 2.34 & -2.34
```

Логические операции также могут использовать скобки. Выполните команды:

```
>> A = 1; B = 0 ;
>> C = ( A & B ) | ~A
```

Дальше логические операции будут использоваться при записи условий и циклов.

## 5. Встроенные и пользовательские функции

Программа MATLAB имеет много встроенных математических функций. Специальные функции из некоторых разделов математики необходимо выбрать при установке MATLAB, но в этом курсе они не используются. Набор таких функций называется библиотекой функций (или просто библиотекой). Например, есть библиотеки для обработки видео в реальном времени, взаимодействия с промышленными устройствами и т.п.

В этом пособии уже были функции `sqrt()` и `log()`. Название функции содержит только латинские буквы и цифры, начинается с буквы. В скобках указывается названия переменной. Если переменных несколько, то они разделяются запятыми. На языке математики такие переменные называются аргументами функции (или просто аргументами). Выполните функцию натурального логарифма `log()`, которая принимает один аргумент, и функцию извлечения корня произвольной степени `nthroot()`, которая принимает два аргумента:

```
>> log ( 2.7183 )
>> nthroot ( 27 , 3 )
```

Результат выполнения функции на языке математики называется значением функции (или просто значением). Если не указана переменная, в которую записывается результат, то используется переменная с названием `ans`

(«answer», ответ). Если в конце команды поставлена точка с запятой, то результат не будет показан в окне команд. Выполните команды с указанием переменной для записи результата.

```
>> a = log ( 2.7183 )  
>> a = log ( 2.7183 );
```

Некоторые функции дают в результате не одно число, а несколько. В этом случае переменная *ans* будет равна только первому числу по порядку в результате. Чтобы получить все числа, необходимо указать переменные для записи результата в квадратных скобках через запятую. Пример показан далее, после записи своей функции.

Матрица чисел или вектор чисел рассматривается как одно число, т.е. как один аргумент и одно значение. Подробнее работа с матрицами разбирается позднее.

Пользователь может написать свою функцию и сохранить в отдельном файле. Это удобно, если требуется ввести несколько команд подряд, или если функция используется несколько раз.

Для этого в верхней части программы нажмите самую левую кнопку "New Script" ("новый скрипт", т.е. новый перечень команд), как это показано на рисунке 5.1.

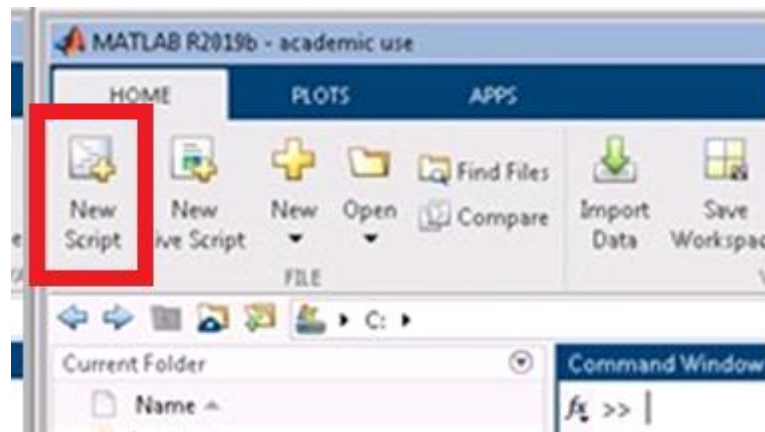


Рисунок 5.1 — Кнопка "новый скрипт"

В центре экрана при этом появляется новое окно "Editor" ("редактор"), с которым можно работать как с обычным текстовым файлом. Окно команд при этом сдвинулось вниз, как показано на рисунке 5.2 Введите в редактор команды, как показано на рисунке:

```
A = 1;  
B = 2;  
C = ( A + B ) * pi;  
disp(C);
```

Чтобы сохранить файл, нажмите сверху экрана кнопку "Save" и введите имя *"my\_script"*.

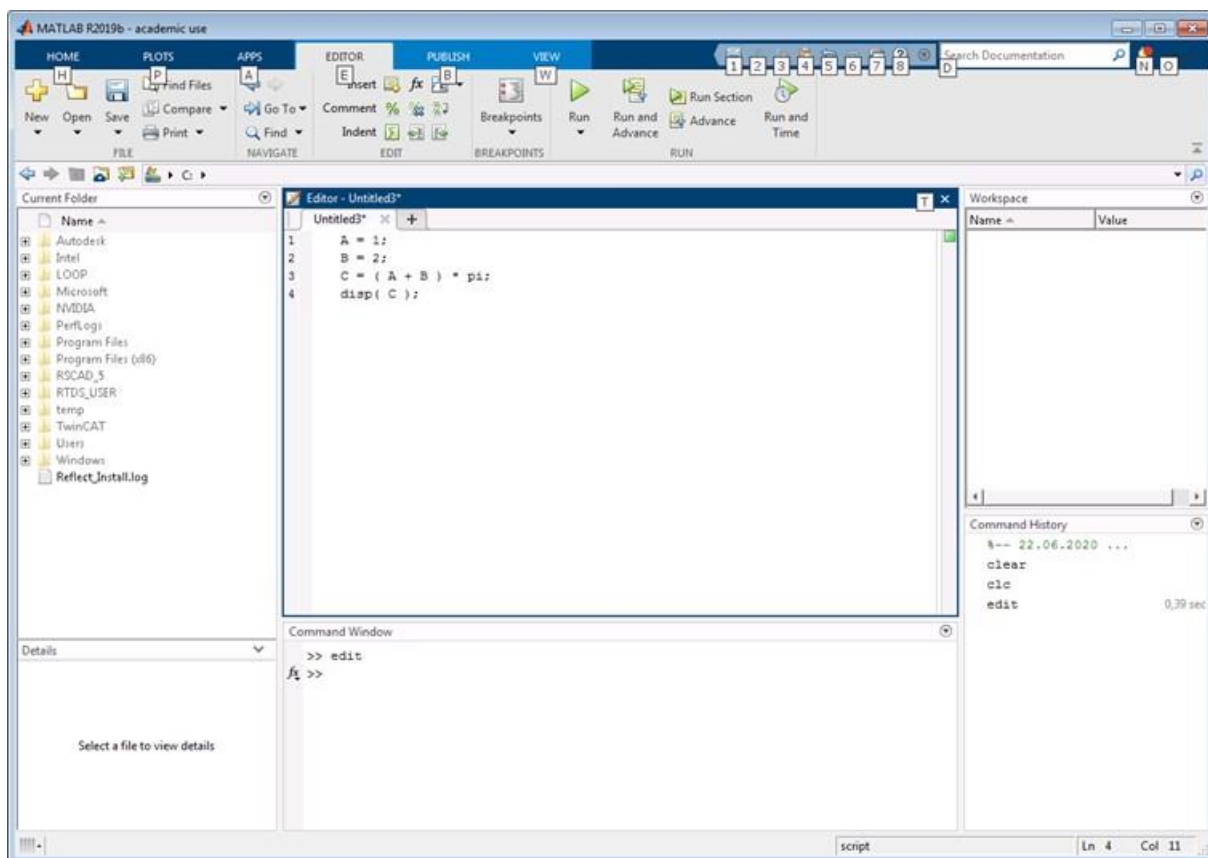


Рисунок 5.2 — Экран редактирования скрипта

Чтобы выполнить команды, нажмите

- на клавиатуре  $F5$  или
- сверху на экране кнопку "Run" или
- в окне команд введите *my\_script*.

Если возникает ошибка

```
>> my_script
```

'my\_script' is not found in the current folder or on the MATLAB path, but exists in: (далее указан путь к файлу)

то это обозначает, что нужно в окне слева "Current folder" открыть папку, в которой находится файл "*my\_script.m*".

Если скрипт выполнен без ошибок, то появится значение переменной *C*, которое выводится на экран с помощью специальной функции *disp()*, т.е. "показать".

Если первая строка файла содержит ключевое слово *function*, то такой файл будет называться *m-функцией*, то есть функцией MATLAB.

Отличие функции от скрипта:

- все переменные, которые созданы внутри функции, удаляются после завершения функции;
- функция может иметь входные и выходные переменные;

- название текстового файла должно строго соответствовать названию функции.

В большинстве случаев лучше использовать функции вместо скриптов, так как функции помогают разбивать программу на небольшие смысловые части, в то время как скрипты просто сохраняют команды в файл.

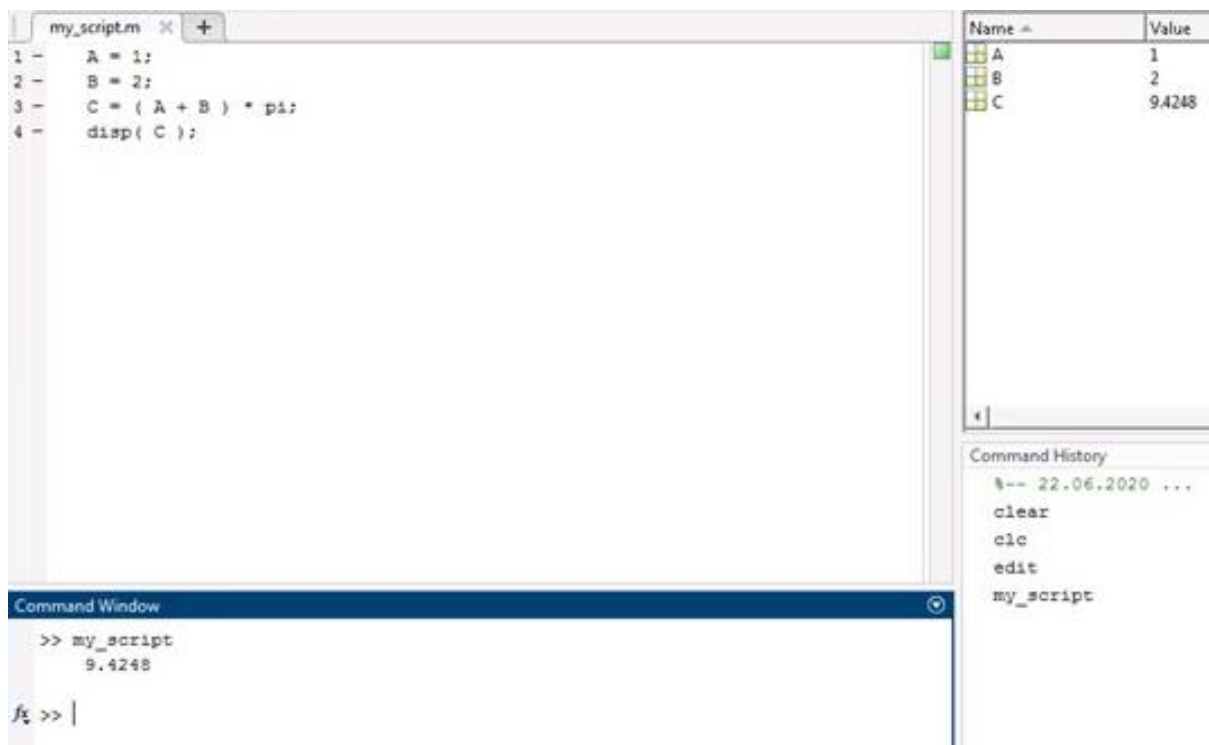


Рисунок 5.3 — Результат выполнения скрипта

Нажмите в верхней части экрана кнопку "New" и введите в редакторе текст функции:

```
function [ C ] = my_function ( A , B )
    C = ( A + B ) * pi;
end
```

Текстовый файл функции обязательно начинается с ключевого слова *function*. После него в квадратных скобках указывается название переменной, которая будет использоваться как результат выполнения функции. Далее идёт любое название функции, которое должно совпадать с названием текстового файла. Далее в круглых скобках через запятую указываются входные переменные (аргументы функции). Пример функции на рисунке 5.4

Сохраните этот файл с названием *my\_function* и затем в окне команд введите:

```
>> my_function( 1 , 2 )
```

Результат аналогичен выполнению скрипта. Также введите команду для записи значения функции в переменную *C*.

```
>> c = my_function( 3 , 4 )
```



```
my_script.m x my_function.m x +
1 function [ C ] = my_function ( A , B )
2
3     C = ( A + B ) * pi;
4
5 end

Command Window
>> my_function( 1 , 2 )

ans =

    9.4248

fx >> |
```

Рисунок 5.4 — Результат выполнения функции

## 6. Матрицы, вектора и массивы

Название программы MATLAB обозначает "*matrix laboratory*", т.е. "лаборатория матриц". В MATLAB активно используется понятие матрицы, т.е. таблицы чисел. Все функции программы MATLAB могут работать с матрицами.

В других языках программирования обычно понятие матрицы и вектора заменяют на понятие "массив", но в любом случае эти понятия обозначают набор чисел, которые идут друг за другом в определённом порядке.

Вектором принято называть матрицу с одним столбцом, т.к. в алгебре принято каждую координату в пространстве записывать в отдельную строку, например:

$\begin{pmatrix} x \\ y \end{pmatrix}$  — точка в двумерном пространстве;

$\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  — точка в трёхмерном пространстве.

Начало вектора находится в точке с координатами 0.

Матрицы в MATLAB записываются в квадратных скобках, внутри одной строки элементы отделяются пробелами, разные строки разделяются точкой с запятой. После квадратной скобки и точки с запятой можно ставить разрыв строки клавишей "Ввод", чтобы матрица хорошо выглядела в файле.

Создайте новый скрипт и напишите в нём следующие команды

```
A = [ 1 2 3 ]
```

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

```
C = [ 1 2 3 ; 4 5 6; 7 8 9]
```

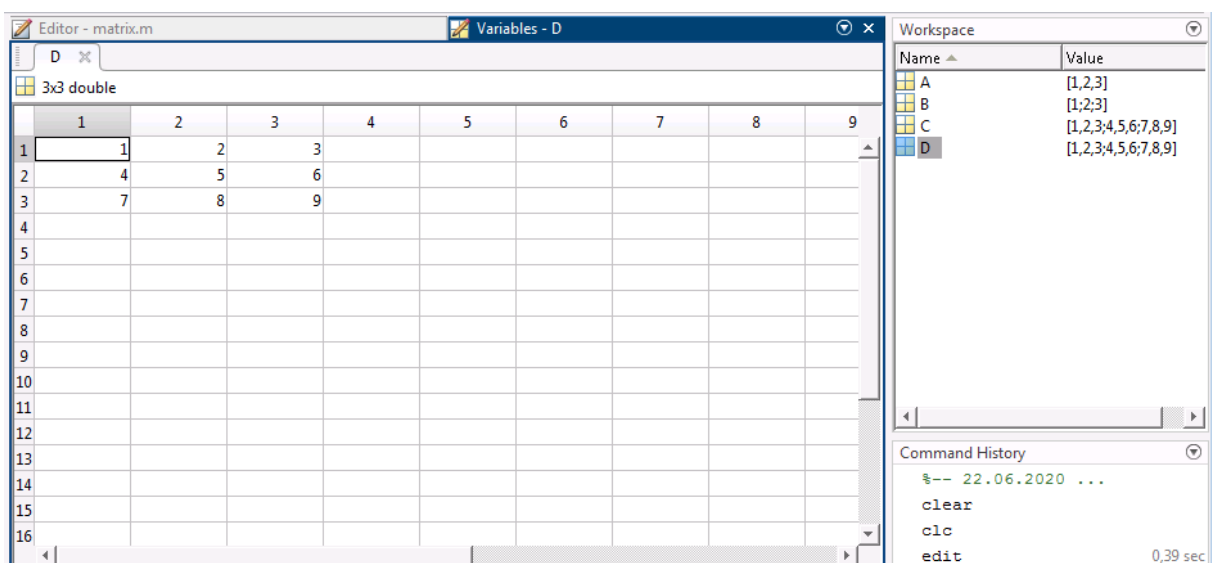
```
D = [  
    1 2 3 ;  
    4 5 6;  
    7 8 9]
```

Сохраните скрипт под названием "*my\_matrix*" и выполните его. Результат выполнения:

```
A =  
    1    2    3  
B =  
    1  
    2  
    3  
C =  
    1    2    3  
    4    5    6  
    7    8    9  
D =  
    1    2    3  
    4    5    6  
    7    8    9
```

Здесь переменная *A* представляет собой строку, переменная *B* - столбец, переменные *C* и *D* - одну и ту же матрицу, но по-разному записанную.

В окне "Workspace" дважды нажмите на название переменной, чтобы увидеть её строки и столбцы, как показано на рисунке 6.1. Нажмите на крестик на синем фоне, чтобы вернуться к редактору.



## Рисунок 6.1 — Значение матрицы

Матрицы подчиняются правилам матричной алгебры:

- сумма матриц равна новой матрице, в которой каждая ячейка равна сумме таких же ячеек слагаемых матриц;
- произведение матрицы на числа равно произведению каждой ячейки на число;
- умножение двух матриц возможно, только если кол-во столбцов в первой матрице равно кол-ву строк во второй матрице;
- единичная матрица имеет по диагонали единицы, а все остальные элементы нули.

Выполните следующие примеры с матрицами  $B$  и  $C$ :

```
>> C + C
>> C * B
>> C * C
>> 3 * B
>> -1 * C
>> 0 * C
```

Выполните умножение матриц неправильно, в результате чего появится ошибка

```
>> B * C
```

Чтобы найти обратную матрицу, введите

```
>> C ^ -1
```

Чтобы транспонировать матрицу, введите

```
>> C'
```

Чтобы не использовать правила матричной алгебры, а вместо этого просто поэлементно умножить или поделить числа в двух матрицах или возвести все числа матрицы в степень, используйте команду с точкой:

```
>> C .* C
>> C ./ C
>> C .^ 2
```

Для того, чтобы вывести на экран только одно число из матрицы, введите название матрицы и индекс элемента в скобках. Индекс записывается так: сначала номер строки, затем столбца. Выполните команды:

```
>> C(1,1)
>> B(1,1) + C(3,3)
```

Попробуйте ввести неправильный индекс, которого нет в матрице, в результате появится ошибка.

```
>> B(3,3)
```

Вместо индекса также можно указать любую переменную, часто используют индексы  $i j k m n$ .

```
>> i = 1;  
>> j = 2;  
>> C(i, j)
```

Если требуется вывести всю строку или столбец, то замените её индекс на двоеточие. Введите команды:

```
>> C(i, :)  
>> C(:, j)  
>> C(:, j) + B
```

Для векторов можно использовать только первый индекс, а номер столбца не указывать, т.к. столбец у вектора всегда один.

```
>> B(1)  
>> B(j)
```

Матрицы можно объединить, если вместо названия столбцов и строк использовать названия других матриц. При этом матрицы должны совпадать друг с другом по кол-ву строк или столбцов в зависимости от того, вертикально или горизонтально они соединяются: Выполните команды и обратите внимание на результат:

```
>> E = [B B B]  
>> F = [C ; C]
```

## 7. Циклы и условия

В программах очень часто используются условия, т.е. сравнение переменных, по результатам которого выполняется заданная команда. В MATLAB для этого используются ключевые слова *if – else – end* ("если – иначе – конец")

Откройте новый файл скрипта и напишите в нём команды:

```
A = 1;  
B = 2;  
if A == B  
    disp('TRUE')  
else  
    disp('FALSE')  
end
```

Сохраните файл и назовите его *"my\_condition"*. Выполните команду и получите результат, показанный на рисунке 7.1

```
>> my_condition  
FALSE
```

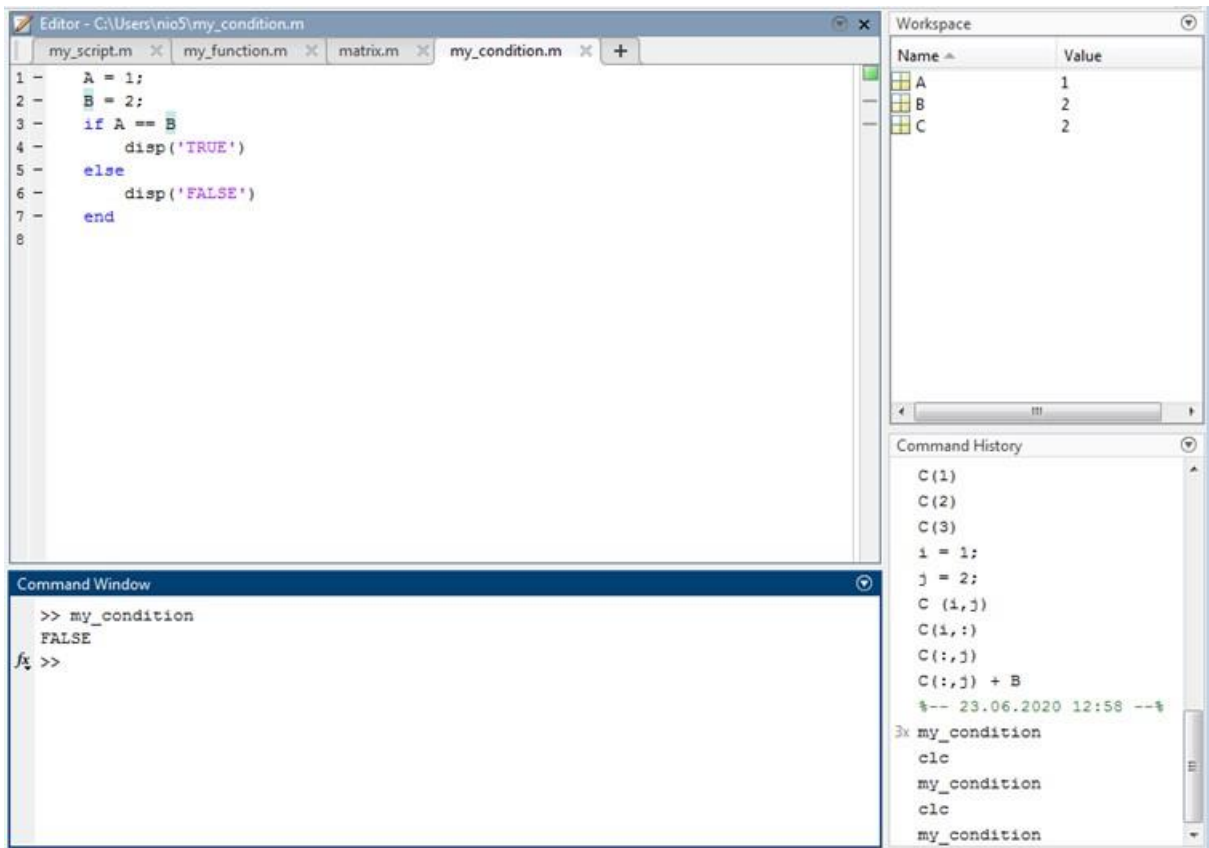


Рисунок 7.1 — Условные операторы

В этом скрипте первая команда *if* сравнивает две переменные *A* и *B*. Обратите внимание, что для сравнения используется два знака "равно", т.к. программе MATLAB нужно отличить операцию "присвоить" от операции "сравнить". Если результат сравнения равен ИСТИНА, то выполняется следующая строчка `disp('TRUE')`. Команды выполняются до тех пор, пока не достигнута строка с ключевым словом *else*.

Если результат сравнения ЛОЖЬ, то программа переходит к ключевому слову *else* и выполняет команды после него до достижения ключевого слова *end*.

Если слово *else* отсутствует, то после результата ЛОЖЬ при выполнении *if* программа сразу переходит к слову *end*.

Вместо слова *else* может быть добавлено ещё одно слово *elseif*, которое добавляет ещё одну проверку оператором *if*. Выполните следующий скрипт:

```

if A == B
    disp('TRUE')
elseif A <= B
    disp('elseif TRUE')
else
    disp('FALSE')
end

```

В программировании также часто используются циклы. Это условия, которые исполняются несколько раз подряд. В программе MATLAB используются два основных цикла: *while* ("пока верно условие") и *for* ("для каждого элемента из множества").

Создайте новый скрипт и напишите в нём команды:

```
a = 0;
for i = 1 : 10
    a = a + 1;
    disp (a);
end
```

Сохраните скрипт с названием *my\_for* и выполните его. В результате будут выведены числа от одного до десяти, т.е. значения переменной *a*.

После ключевого слова *for* записывается название переменной, которая будет использоваться внутри цикла. Очень часто используют букву *i*, а переменную называют "счётчик". Дальше после знака "равно" записывается начальное значение переменной, а после двоеточия - конечное значение. В этом примере на первом шаге цикл будет выполнен со значением  $i = 1$ , а на последнем шаге  $i = 10$ . При каждом выполнении цикла (шаг) переменная *i* увеличивается на 1. Внутри цикла пишутся команды, а цикл повторяется при достижении ключевого слова *end*. В цикле *for* значение переменной увеличивается автоматически.

Если в цикле *for* требуется увеличивать счётчик не на единицу, а на другое число (например, двойку), то первую строку цикла надо записать так

```
for i = 1 : 2 : 10
```

Команды внутри условия и внутри цикла принято "отбивать" клавишей табуляции или четырьмя пробелами, т.е. делать увеличенный отступ.

Далее рассмотрен цикл *while*. Создайте новый скрипт и назовите его *my\_while*. Напишите и выполните следующие команды:

```
a = 0;
i = 1;
while i <= 10
    a = a + 1;
    disp (a) ;
    i = i + 1;
end
```

Результат выполнения этого цикла такой же, как в цикле *for*. Для работы с циклом *while* необходимо задать начальное значение переменной-счётчика, это делается строчкой  $i = 1$ .

После ключевого слова *while* записывается условие в виде логического выражения. Пока это условие верно (ИСТИНА), выполняются команды до ключевого слова *end*. Внутри цикла необходимо увеличить значение переменной-счётчика, чтобы цикл не выполнялся бесконечно.

Относительно нечасто внутри циклов используются ключевые слова *break* и *continue*. Ключевое слово *break* принудительно заканчивает выполнение цикла и переходит к команде после слова *end*, а слово *continue* пропускает текущий шаг цикла и снова возвращается к началу цикла. Выполните следующую команду:

```
for i = 1 : 10
    disp (a);
    if i = 5
        break;
    end
end
```

## 8. Вывод графиков

В программе MATLAB можно построить все возможные виды графиков: двумерные, трёхмерные, цветные, диаграммы со столбцами или с кружками. Все графики можно оформить по своему желанию, добавить произвольные надписи. С другой стороны, это делается с помощью команд, поэтому оформление каждого графика может потребовать много усилий. Здесь рассмотрим только базовые принципы.

Задайте строку с числами в диапазоне от 0 до 10 с шагом 0.1 с помощью команды:

```
t = 0 : 0.1 : 10;
```

Это будут значения по оси времени на графике. Теперь по формуле синуса рассчитаем значения функции  $y(t)$ :

```
y = sin(2*pi*t);
```

Чтобы вывести результаты на график, введите:

```
plot(t, y);
```

Эта функция открывает новое окно с названием "Figure", на котором показан график. Масштаб на этом графике подбирается так, чтобы все значения уместились на график.

У функции *plot* первый аргумент - значения по оси  $x$ , второй аргумент - значения по оси  $y$ . Эти матрицы должны содержать одинаковое количество элементов, иначе программа выдаст ошибку.

Добавьте на график ещё одну величину косинуса:

```
z = cos(2*pi*t);
```

Теперь перепишите функцию *plot* так, чтобы на рисунке были показаны два графика:

```
plot(t, y, '-r', t, z, '--b');
```

Здесь после значения по оси *x* и *y* добавлены тип линии и цвет линии. Они обязательны и указываются в одинарных кавычках (апостроф). Символ обозначает тип линии:

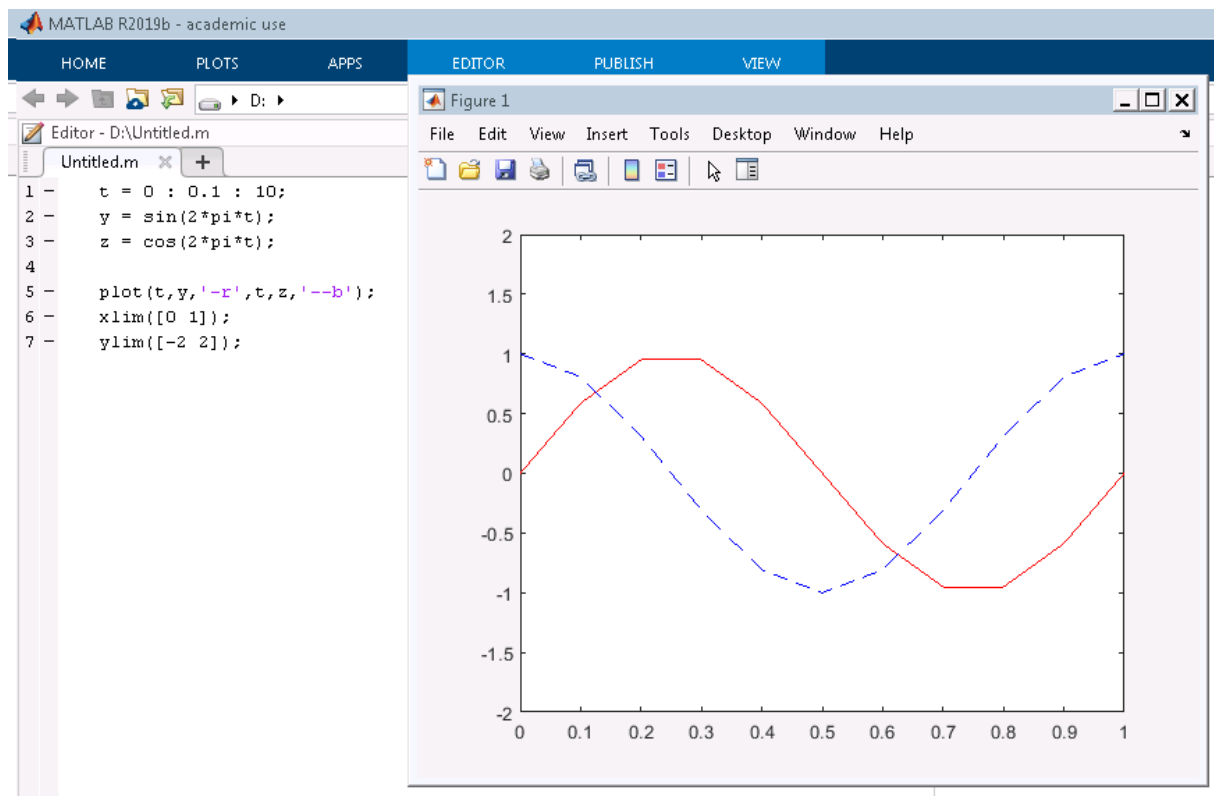
- одно тире - сплошная линия;
- два тире - пунктирная линия;
- двоеточие - точечная линия;
- тире точка - пунктирная линия с точками
- звёздочка - линия со звёздочками.

Буквой обозначается цвет по-английски: *red*, *blue*, *green*, *black*, *magenta*.

Если требуется задать максимум и минимум по осям, то ПОСЛЕ функции *plot* введите команды:

```
xlim([0 1]);  
ylim([-2 2]);
```

Здесь в скобках укажите границы по оси *x* и *y* соответственно



Если при выводе на график результаты расчётов накладываются друг на друга, то введите команду *clf*, чтобы очистить график ("clear figure", очистить рисунок).

Чтобы вывести несколько графиков по вертикали или горизонтали, добавьте перед командой *plot* другую команду:



```
subplot(2,1,1)
plot(t,y);
subplot(2,1,2)
plot(t,z);
```

В функции *subplot* первое значение - количество рисунков по вертикали, второе значение - количество рисунков по горизонтали, третье - номер рисунка, на который вы собираетесь выводить свой график с помощью *plot*.

Если требуется более сложное оформление графики, то смотрите руководство к функции *plot*.

## 9. Расчётное задание

### 9.1. Решение квадратного уравнения

Решим простое квадратное уравнение с помощью MATLAB. Для уравнения вида

$$ax^2 + bx + c = 0$$

нам даны коэффициенты  $a = -1$ ,  $b = 2$ ,  $c = 3$ . Решение такого уравнения  $x_1 = -1$ ,  $x_2 = 3$ .

Для этого наберите команды:

```
function [x1,x2] = lab1( )
    a = -1;
    b = 2;
    c = 3;
    D = b^2 - 4*a*c;
    x1 = ( -b + sqrt(D) ) / ( 2 * a );
    x2 = ( -b - sqrt(D) ) / ( 2 * a );
end
```

Сохраните файл под таким же именем *lab1.m*, что и название функции. Чтобы получить результат, введите в главном окне команд

```
[x1 , x2] = lab1()
```

Также можно просто убрать точки с запятой в строчках с  $x_1$  и  $x_2$ , чтобы увидеть ответ.

### 9.2. Вычислить логическое выражение

Решим логическое выражение, заданное в виде

$$( \text{НЕ } a \text{ И } b ) \text{ И } ( c < d ) \text{ ИЛИ } ( e \text{ И НЕ } f )$$

при  $a = 1, b = 0, c = 1, d = 0, e = 1, f = 0$ .

$(\text{НЕ } 1 \text{ И } 0) \text{ И } (1 < 0) \text{ ИЛИ } (1 \text{ И НЕ } 0)$

Решим отдельно по скобкам:

Решим отдельно по скобкам:

$(\text{НЕ } 1 \text{ И } 0) = (\text{НЕ ПРАВДА И ЛОЖЬ}) = (0 \text{ И } 0) = 0 = \text{ЛОЖЬ}$

$(1 < 0) = 0 = \text{ЛОЖЬ}$

$(1 \text{ И НЕ } 0) = (\text{ПРАВДА И НЕ ЛОЖЬ}) = (1 \text{ И } 1) = 1 = \text{ПРАВДА}$

Объединим эти решения разных скобок:

$0 \text{ И } 0 \text{ ИЛИ } 1 = 0 \cdot 0 + 1 = 1 = \text{ПРАВДА}$

Для решения в программе запишем выражение:

```
function [ x ] = lab2( )
    a = 1; b = 0 ; c = 1; d = 0; e = 1; f = 0;
    ab = ~a & b;
    cd = c < d;
    ef = e & ~f;
    x = ab & cd | ef;
end
```

Результат записан в переменную x.

### 9.3. Работа с массивом

Найдём для массива значения суммы и произведения всех членов.  
Даны элементы массива:

$[-1; -2; 4; 7; 3; 1; 2; 4; -3; 2]$

Чтобы найти эти величины, запишите команды:

```
function [ ] = lab3( )
    A = [ -1 ; -2 ; 4 ; 7 ; 3 ; 1 ; 2 ; 4 ; -3 ; 2 ];
    summ = 0; %в этой переменной будет сумма
    prod = 1; %в этой - произведение
    for i = 1 : length(A)
        summ = summ + A(i);
        prod = prod * A(i);
    end
    disp('Сумма всех элементов:');
    disp(summ);
```

```

disp('Произведение всех элементов:');
disp(prod);
end

```

Результат выполнения будет такой:

```

Сумма всех элементов:
    17
Произведение всех элементов:
   -8064

```

#### 9.4. Работа с комплексными числами

Пусть даны четыре комплексных числа в показательном виде  $Ae^{j\varphi}$ . Найдём эти числа в алгебраическом виде  $Re + jIm$  и построим на графике синусоиды, заданные этими числами. Частота синусоид равна 50 Гц, построить для одного периода.

Даны числа:

$$\dot{A}_1 = 0.5e^{j30^\circ}; \dot{A}_2 = -0.5e^{j90^\circ}; \dot{A}_3 = 0.8e^{j0^\circ}; \dot{A}_4 = -0.2e^{-j20^\circ}$$

В MATLAB углы записываются в радианах, а не в градусах, поэтому угол нужно поделить на 180 градусов и умножить на число пи.

Чтобы найти все нужные значения, запишем команды:

```

function [ ] = lab4( )
    A1 = 0.5*exp(1i*pi/6);
    A2 = -0.5*exp(1i*pi/2);
    A3 = 0.8*exp( 0 );
    A4 = -0.2*exp(-1i*20*pi/180);
    w = 2*pi*50;
    t = 0:0.001:0.02;
    a1 = abs(A1) * sin( w*t + atan(imag(A1)/real(A1)) );
    a2 = abs(A2) * sin( w*t + atan(imag(A2)/real(A2)) );
    a3 = abs(A3) * sin( w*t + atan(imag(A3)/real(A3)) );
    a4 = abs(A4) * sin( w*t + atan(imag(A4)/real(A4)) );
    plot(t,a1,'-g',...
         t,a2,'--r',...
         t,a3,'-.b',...
         t,a4,'.k');
end

```

## 10. Варианты заданий

Вариант	Задание 1			Задание 2						Задание 3				Задание 4							
	a	b	c	формула						массив				A1	A2	A3	A4	ψ1	ψ2	ψ3	ψ4
1	-9	-48	-49	$(a > \text{НЕ } b) < (c \text{ И } d) \text{ ИЛИ } (e \geq \text{НЕ } f)$						[-2; 8; 4; -1; -4; -1; -1; -8; -6; 2]				0,25	0,51	0,86	0,36	-94	75	74	43
2	-8	-69	-12	$(a \geq b) < (c \text{ И НЕ } d) \text{ ИЛИ } (e \leq \text{НЕ } f)$						[5; 0; 1; 6; -2; -9; -6; -8; 3; -6]				0,58	0,04	0,06	0,8	-121	35	-1	107
3	53	6	-27	$(a \text{ И } b) \text{ ИЛИ } (c \text{ И } d) > (e \text{ И } f)$						[-5; -2; -8; 7; 3; -2; -9; 0; 2; 8]				0,32	0,89	0,41	0,02	-168	168	7	39
4	1	17	-17	$(a \text{ ИЛИ НЕ } b) \geq (c < d) \text{ И } (e \text{ И НЕ } f)$						[8; 6; 0; 3; 1; -8; -4; -3; 7; 8]				0,91	0,3	0,31	0,61	-88	-118	-132	150
5	73	87	-53	$(a \text{ ИЛИ НЕ } b) \leq (c \text{ ИЛИ НЕ } d) < (e \text{ И } f)$						[2; 5; 7; 0; 3; -7; -1; -9; 1; 4]				0,46	0,52	0,78	0,41	94	162	31	154
6	-4	33	31	$(a \text{ ИЛИ НЕ } b) < (c \text{ ИЛИ НЕ } d) \text{ И } (e \text{ И } f)$						[-3; 4; -7; -2; 9; -8; 8; -5; 1; -7]				0,2	0,63	0,15	0,22	-10	50	-180	121
7	-21	-25	15	$(a \geq b) > (c \geq \text{НЕ } d) \geq (e > f)$						[6; 3; 9; 1; 6; 4; 0; -3; -4; 8]				0,79	0,6	0,76	0,47	-139	144	4	159
8	-81	33	85	$(a \text{ И НЕ } b) \text{ И } (c < \text{НЕ } d) \text{ И } (e < f)$						[2; 6; 5; -7; -7; 0; -7; -7; 6; 1]				0,91	0,01	0,92	0,99	-24	-104	-22	75
9	-12	-25	96	$(a \text{ И НЕ } b) < (c \text{ И НЕ } d) \text{ ИЛИ } (e \text{ И НЕ } f)$						[7; 6; -8; 4; -4; 8; -7; -9; 6; 1]				1	0,7	0,18	0,12	-145	141	108	48
10	-70	8	9	$(a \text{ И } b) \text{ И } (c \text{ И } d) \text{ И } (e \geq \text{НЕ } f)$						[-8; -2; -4; -2; 1; -2; -9; 9; -2; 1]				0,55	0,08	0,59	0,07	-92	104	148	29
11	41	-5	-73	$(a \text{ И НЕ } b) \text{ И } (c < d) \text{ И } (e < f)$						[-1; 2; -7; 0; -8; 5; 0; 1; -4; -9]				0,92	0,11	0,88	0,99	-135	-42	-112	-124
12	97	-64	8	$(a \leq b) \text{ ИЛИ } (c \text{ И НЕ } d) \text{ И } (e \text{ И НЕ } f)$						[0; 8; -9; -1; -2; -1; 2; -8; -6; 3]				0,9	0,28	0,23	0,14	136	146	91	97
13	-99	-92	36	$(a \text{ ИЛИ } b) \text{ ИЛИ } (c \leq \text{НЕ } d) \geq (e \text{ ИЛИ НЕ } f)$						[4; -3; 7; 1; -1; 4; 9; -6; -5; 7]				0,08	0,97	0,5	0,28	165	43	31	76
14	74	68	-69	$(a \geq \text{НЕ } b) \text{ И } (c > d) \leq (e \text{ ИЛИ } f)$						[-2; -7; 8; -6; -3; -7; -6; -5; 5; -8]				0,66	0,75	0,1	0,04	154	130	-174	93
15	-21	15	97	$(a \text{ И } b) \text{ И } (c \leq \text{НЕ } d) \text{ И } (e \text{ И } f)$						[7; -4; -5; 0; -4; -2; 3; -6; 2; -7]				0,71	0,87	0,31	0,74	-168	39	85	-13
16	-84	-43	-1	$(a \geq b) \text{ И } (c \text{ И НЕ } d) \leq (e < f)$						[6; 3; 2; 1; -3; 3; 2; 9; -1; -3]				0,38	0,36	0,73	0,48	-144	170	-128	124
17	-94	-96	68	$(a \text{ И } b) \text{ И } (c \geq d) \text{ И } (e \text{ ИЛИ } f)$						[5; 7; -8; -8; -4; -1; 2; -3; 2; -7]				0,55	0,18	0,58	0,32	-57	110	-169	-131
18	-90	5	73	$(a \text{ ИЛИ } b) \text{ И } (c \text{ И } d) \text{ И } (e \text{ И } f)$						[7; 4; 2; -5; 9; 9; 4; -7; 6; -7]				0,72	0,13	1	0,77	-112	133	-62	47
19	72	90	-9	$(a \text{ ИЛИ НЕ } b) \text{ И } (c \leq \text{НЕ } d) \text{ И } (e \text{ И } f)$						[1; -9; -9; -5; -5; -3; -6; 7; 2; 3]				0,11	0,94	0,1	0,05	74	-81	-127	142
20	-79	43	24	$(a \leq b) < (c \text{ ИЛИ НЕ } d) \text{ И } (e > \text{НЕ } f)$						[-4; -4; -5; -2; -8; 6; 1; -7; -5; -7]				0,42	0,14	0,7	0,95	-42	-116	39	-39
21	-36	45	-11	$(a \geq b) \text{ И } (c \text{ И } d) \leq (e > \text{НЕ } f)$						[0; 3; 2; 6; -9; 3; 1; -7; -9; 1]				0,86	0,95	0,66	0,55	-161	-58	8	151
22	-42	84	83	$(a \text{ ИЛИ НЕ } b) \text{ И } (c \text{ И } d) > (e \text{ И } f)$						[-8; -1; -8; 2; -1; -1; -3; -3; -1; 6]				0,6	0	0,89	0,95	79	-35	159	-145
23	-62	2	42	$(a \text{ И НЕ } b) \geq (c > \text{НЕ } d) \text{ И } (e \text{ И } f)$						[-6; -7; -9; 2; 0; -3; 8; 7; 2; 5]				0,92	0,6	0,29	0,02	6	12	45	136
24	34	-28	-67	$(a > \text{НЕ } b) \text{ И } (c \text{ И НЕ } d) \geq (e < f)$						[3; 2; 1; 1; -9; -3; -4; -4; -9; 6]				0,81	0,75	0,69	0,65	-126	55	-143	-87
25	74	-48	-81	$(a \text{ И } b) \text{ ИЛИ } (c \text{ И } d) \text{ И } (e \text{ И } f)$						[-1; 9; -4; -3; 6; 7; -3; 7; 0; -5]				0,57	0,33	0,13	0,65	17	81	-67	-157

Вариант	Задание 1			Задание 2						Задание 3				Задание 4							
	a	b	c	формула						массив				A1	A2	A3	A4	ψ1	ψ2	ψ3	ψ4
26	-8	-26	12	(a И НЕ b) И (c И d) <= (e И НЕ f)						[ 6 ; -2 ; -3 ; -2 ; 3 ; 7 ; -9 ; -2 ; 6 ; 0 ]				0,12	0,05	0,27	0,51	28	-35	27	-163
27	-62	6	58	(a >= НЕ b) <= (c <= d) >= (e >= НЕ f)						[ -6 ; -3 ; -3 ; 9 ; 0 ; 0 ; 1 ; -5 ; 0 ; 5 ]				0,72	0,14	0,97	0,42	161	-73	-126	-86
28	-46	-19	5	(a ИЛИ b) И (c >= НЕ d) >= (e И f)						[ 7 ; 0 ; -3 ; -3 ; -9 ; 6 ; -1 ; 1 ; 3 ; -2 ]				0,96	0,97	0,54	0,83	89	-177	-107	174
29	-89	36	24	(a ИЛИ НЕ b) И (c И d) <= (e И НЕ f)						[ 1 ; 2 ; -7 ; -5 ; 1 ; 8 ; 2 ; -2 ; 4 ; 8 ]				0,15	0,87	0,8	0,2	56	5	179	-93
30	11	-30	-55	(a И НЕ b) И (c <= НЕ d) >= (e > f)						[ -8 ; 4 ; 2 ; 6 ; 0 ; 3 ; 8 ; 6 ; 8 ; 4 ]				0,82	0,17	0,79	0,63	-164	1	69	168
31	-29	25	61	(a И b) >= (c И НЕ d) < (e >= НЕ f)						[ 6 ; -9 ; 7 ; -8 ; -5 ; -5 ; 4 ; -7 ; -4 ; -8 ]				0,8	0,14	0,13	0,56	-101	-110	33	127
32	-89	-79	94	(a И НЕ b) ИЛИ (c >= НЕ d) < (e >= f)						[ -8 ; -6 ; 9 ; 6 ; -6 ; 1 ; 5 ; -6 ; -8 ; 1 ]				0,12	0,06	0,38	0,75	-38	149	-133	-96
33	9	-52	-59	(a И НЕ b) < (c >= НЕ d) >= (e > НЕ f)						[ -4 ; -7 ; -6 ; 4 ; 9 ; 1 ; -4 ; -8 ; 9 ; -4 ]				0,62	0,71	0,02	0,9	-24	-80	-16	123
34	-71	-63	-4	(a ИЛИ НЕ b) >= (c < НЕ d) >= (e И НЕ f)						[ -8 ; -9 ; 5 ; 6 ; 9 ; -1 ; 1 ; 6 ; 1 ; 4 ]				0,59	0,53	0,32	0,82	160	-168	-11	50
35	62	20	-43	(a И НЕ b) ИЛИ (c >= НЕ d) И (e <= f)						[ -7 ; 7 ; 1 ; -4 ; 5 ; 2 ; 4 ; 7 ; -8 ; -7 ]				0	0,61	0,72	0,7	-98	116	97	72
36	7	6	-45	(a И b) И (c ИЛИ НЕ d) И (e И f)						[ 4 ; 6 ; -9 ; 4 ; 2 ; 9 ; -4 ; -4 ; 7 ; 2 ]				0,34	0,83	1	0,53	143	88	-126	168
37	2	35	75	(a >= b) И (c И НЕ d) И (e >= f)						[ 8 ; 6 ; -6 ; 4 ; 6 ; 5 ; 5 ; -8 ; -1 ; -5 ]				0,19	0,01	0,89	0,45	-31	19	132	6
38	-44	25	71	(a И b) ИЛИ (c И НЕ d) И (e И f)						[ 4 ; 9 ; 1 ; -2 ; 4 ; -6 ; 7 ; 1 ; 2 ; 3 ]				0,36	0,65	0,05	0,61	92	-133	-81	97
39	66	76	-55	(a < НЕ b) И (c < НЕ d) ИЛИ (e < f)						[ 9 ; -4 ; -9 ; -7 ; -8 ; 8 ; -4 ; -7 ; 6 ; 3 ]				0,9	0,63	0,77	0,2	-171	150	-116	100
40	80	10	-84	(a <= b) И (c И НЕ d) ИЛИ (e И f)						[ -7 ; -3 ; -6 ; -8 ; 2 ; 5 ; 0 ; -6 ; -9 ; -9 ]				0,85	0,28	0,63	0,82	51	-172	-130	-167
41	-91	95	59	(a >= НЕ b) ИЛИ (c >= d) И (e И НЕ f)						[ -2 ; -4 ; 6 ; 8 ; 3 ; -9 ; 2 ; -2 ; 9 ; 3 ]				0,11	0,84	0,7	0,63	-46	-59	-62	-94
42	-40	7	97	(a ИЛИ НЕ b) < (c > НЕ d) >= (e И НЕ f)						[ -3 ; -9 ; 1 ; 4 ; -4 ; 3 ; 0 ; 2 ; 9 ; 2 ]				0,24	0,97	0,55	0,25	-149	-73	169	11
43	-81	-59	81	(a ИЛИ НЕ b) > (c И d) И (e И НЕ f)						[ 6 ; 6 ; -9 ; -4 ; -2 ; -6 ; -7 ; -1 ; -5 ; 4 ]				0,78	0,11	0,67	0,7	89	-7	-143	119
44	64	-48	-92	(a <= b) >= (c И d) И (e > НЕ f)						[ 4 ; 0 ; -1 ; 0 ; -2 ; -6 ; -8 ; 3 ; 3 ; 6 ]				0,58	0,33	0,14	0,37	4	9	61	-32
45	6	80	-66	(a И НЕ b) ИЛИ (c ИЛИ НЕ d) И (e <= НЕ f)						[ 4 ; 9 ; 4 ; -7 ; 2 ; -6 ; 0 ; -5 ; -5 ; -2 ]				0,17	0,73	0,82	0,52	-60	1	-102	160
46	93	-26	-84	(a >= b) И (c > d) <= (e И f)						[ 0 ; 4 ; 2 ; 7 ; 9 ; -9 ; 9 ; 9 ; 6 ; 3 ]				0,41	0,94	0,09	0,17	-36	-131	112	34
47	-14	91	-83	(a И b) < (c >= d) ИЛИ (e И f)						[ -2 ; -6 ; -9 ; 3 ; 1 ; 6 ; -9 ; -5 ; -7 ; 3 ]				0,25	0,84	0,9	0,93	34	-170	-161	113
48	-14	-87	74	(a И НЕ b) И (c И НЕ d) >= (e > НЕ f)						[ 4 ; -2 ; 0 ; 2 ; 9 ; 3 ; 3 ; 3 ; -1 ; 6 ]				0,42	0,73	0,79	0,55	161	-20	46	175
49	-60	-57	33	(a < b) <= (c И d) ИЛИ (e > НЕ f)						[ 4 ; 0 ; 4 ; 3 ; -4 ; -7 ; -5 ; -7 ; 4 ; -2 ]				0,91	0,67	0,9	0,69	-126	-36	-49	-42
50	-41	44	70	(a >= b) ИЛИ (c И НЕ d) И (e И НЕ f)						[ 2 ; -4 ; 1 ; -1 ; 5 ; 0 ; 7 ; 8 ; -1 ; 4 ]				0,92	0,07	0,42	0,28	133	-121	-85	179
51	-94	23	52	(a И НЕ b) ИЛИ (c ИЛИ НЕ d) И (e > НЕ f)						[ -5 ; -1 ; 3 ; -4 ; 1 ; 6 ; -4 ; -7 ; 7 ; 0 ]				0,26	0,45	0,3	0,05	178	121	-10	41
52	-70	-9	83	(a ИЛИ b) <= (c > НЕ d) И (e И НЕ f)						[ -6 ; 8 ; -3 ; 5 ; 5 ; -2 ; 1 ; 1 ; 1 ; -9 ]				0,37	0,09	0,82	0,83	-144	19	-57	-96

Вариант	Задание 1			Задание 2						Задание 3				Задание 4							
	a	b	c	формула						массив				A1	A2	A3	A4	ψ1	ψ2	ψ3	ψ4
53	70	-62	2	( a >= НЕ b ) ИЛИ ( c И d ) > ( e И НЕ f )						[ -9 ; 9 ; 7 ; 2 ; -6 ; -4 ; 2 ; -3 ; -5 ; -7 ]				0,02	0,82	0,16	0,92	148	-168	64	86
54	97	-67	-61	( a ИЛИ НЕ b ) >= ( c И НЕ d ) ИЛИ ( e И НЕ f )						[ -4 ; 8 ; 2 ; 3 ; -3 ; -5 ; -6 ; 5 ; 1 ; 0 ]				0,91	0,3	0,63	0,28	46	16	10	-98
55	14	50	-46	( a И b ) > ( c И d ) > ( e ИЛИ НЕ f )						[ -2 ; 1 ; 3 ; 9 ; -2 ; -1 ; -1 ; 9 ; 8 ; 7 ]				0,94	0,04	0,45	0,75	62	-3	114	-135
56	-88	2	18	( a И НЕ b ) И ( c И НЕ d ) ИЛИ ( e > НЕ f )						[ -2 ; -3 ; -9 ; -2 ; -6 ; -7 ; -6 ; 9 ; -2 ; -8 ]				0,79	0,25	0,41	0,86	139	-42	85	117