# Лабораторная работа №2

# Программирование в среде MATLAB

**Цель работы**: изучение возможностей MATLAB по составлению файлов-сценариев и файлов-функций, в том числе для организации разветвляющихся и циклических программ.

## Рабочее задание

Работа в режиме калькулятора в среде MATLAB не позволяет выполнять циклические алгоритмы вычислений. Такие вычисления удобнее выполнять в виде программ. Для этого следует набирать команды в окне редактирования **Editor**, а затем выполнять их все сразу или частями, в том числе и из командной строки, а также сохранить их в файле с расширением .m и использовать в дальнейшем. Кроме того, в окне редактирования удобнее выполнять отладку программы.

Файлы, которые содержат в себе коды программы MATLAB, называются *M*-файлами. Они бывают двух типов: *файлы-сценарии* (**Script M-Files**) и *файлы-функции* (**Function M-Files**), описывающие функции, определяемые пользователем.

Основные программы, управляющие от начала до конца порядком всего вычислительного процесса, являются файлами-сценариями. Файлы-функции используют для оформления отдельных процедур и функций, при выполнении которых либо внутри файлов-сценариев, либо внутри других файлов-функций необходимо сначала задать значения их входных переменных.

1 Запустите программу MATLAB и настройте имя директории по умолчанию для хранения создаваемых файлов. Для этого выберите команду **Home⇒Set Path⇒Add Folder** (для интерфейса с меню – **File⇒Set Path**…⇒**Add Folder**). В появившемся окне (рисунок 4.1) укажите свою папку, находящуюся на диске H:, нажмите на клавишу Save и закройте окно, подтвердив запрос об использовании указанной папки для хранения создаваемых файлов по умолчанию.

***Обратите внимание, что имя вашей папки должно содержать лишь латинские буквы или цифры без пробелов!***

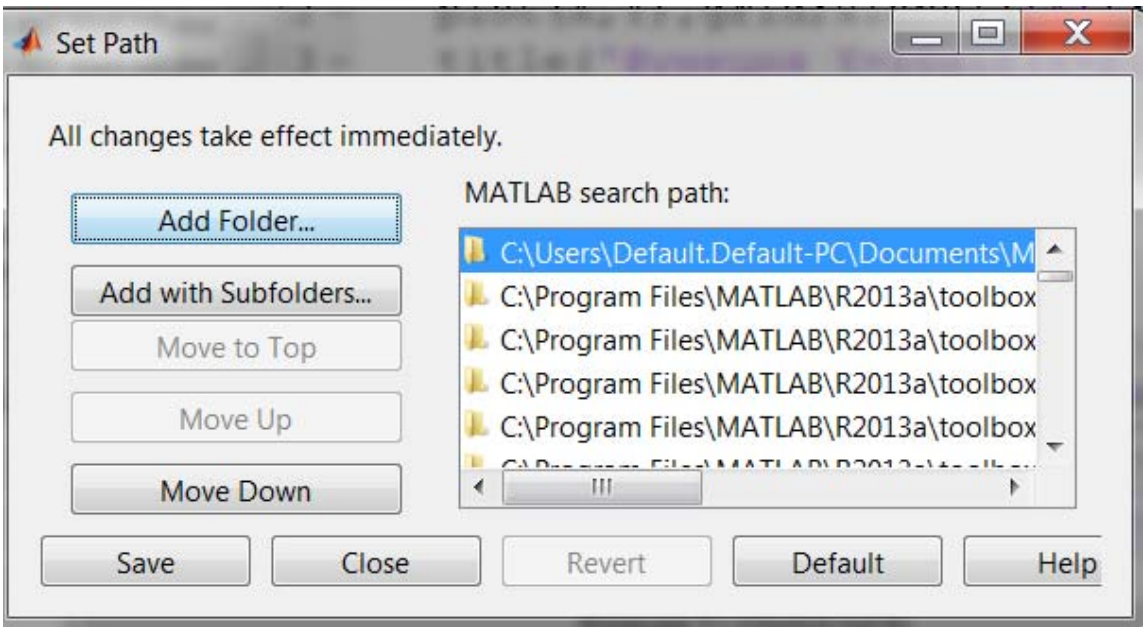


Рисунок 4.1

2 Создайте *M*-файл с помощью команды **Home⇒New⇒Script** (для интерфейса с меню – **File⇒New⇒M-file**) из основного окна MATLAB. В результате откроется окно редактора *M*-файлов **Editor**, в котором наберите команды для построения графика (рисунок 4.2).

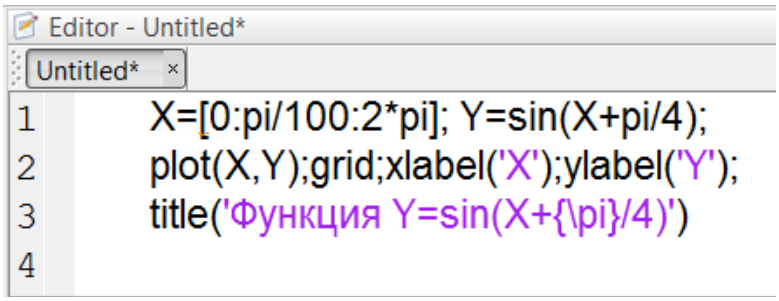


Рисунок 4.2

3 Сохраните *M*-файл в своей папке на диске H: под именем grafik1.m с помощью команды **Editor⇒Save As** (для интерфейса с меню – **File⇒Save As** в окне редактора *M*-файлов).

***Обратите внимание, что имя файла должно содержать лишь латинские буквы или цифры без пробелов!***

4 С помощью команды **Editor⇒Run** (для интерфейса с меню – **Debug⇒Run** в окне редактора *M*-файлов) или путем нажатия клави-

ши F5 выполните все команды из окна редактирования. В результате на экране появится графическое окно **Figure l** с графиком функции и заголовком.

5 Увеличьте в программе диапазон изменения аргумента X в два раза и запустите все команды снова. Посмотрите, как изменится график, и сделайте вывод.

6 Выделите с помощью мыши первые две строки программы в окне редактирования *M*-файлов, нажмите правую клавишу мыши и выберите в появившемся контекстном меню команду **Evaluate Selection** (или нажмите клавишу F9). Обратите внимание, что в графическом окне выведется только график без заголовка.

*Отчет*

Включить в отчет по лабораторной работе содержимое окон редактирования *M*-файлов и графики, полученные при выполнении пп. 2–6.

7 Закройте окно редактирования *M*-файлов и графическое окно. Откройте снова файл **figura1.m** с помощью команды **Home⇒Open** (для интерфейса с меню – **File⇒Open**).

8 В редакторе *M*-файлов может быть одновременно открыто несколько файлов. Переход между файлами осуществляется с помощью закладок с именами файлов. Создайте новый *M*-файл под именем **fazagrad.m** как файл-функцию и определите в нем новую *M*-функцию с помощью ключевого слова **function**. Формат заголовка *M*-функции имеет следующий вид:

**Function [выходные переменные] = Имя (входные переменные)**

Определите в окне редактора *M*-файлов **Editor** (рисунок 4.3) функцию под именем **fazagrad**, которая предназначена для перевода фазового сдвига между напряжением и током на участке электрической цепи, измеренного по осциллографу в секундах (**fazac**), в градусы (**fazag**).

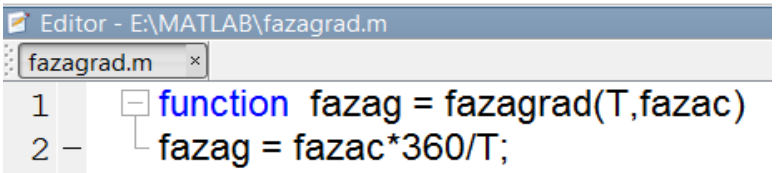


Рисунок 4.3

В качестве входных параметров данной функции выступают измеренные по осциллографу (в секундах) период гармонического сигнала *T* и фазовый сдвиг **fazac**. Данная функция возвращает свое значение (в градусах) переменной **fazag**.

9 Сохраните созданную *M*-функцию с помощью команды **File⇒Save** и закройте окно редактирования.

10 Обращение к созданной функции **fazag** может быть выполнено или в командном окне, или в окне редактора файла-сценария. На рисунке 4.4 представлен пример обращения к функции **fazagrad** из командного окна.

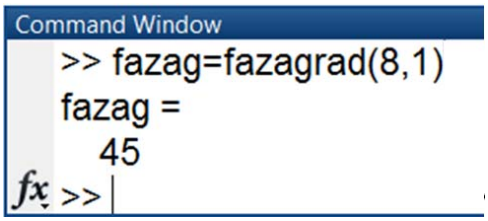


Рисунок 4.4

*Отчет*

Включить в отчет по лабораторной работе содержимое окна редактирования, полученное при выполнении п. 8, а также результаты обращения к функции для трех различных значений ее аргументов.

11 Как в файлах-сценариях, так и в файлах-функциях для организации разветвляющиеся и циклические вычисления используют такие операторы управления вычислительным процессом, которые начинаются со служебных слов **if**, **while** или **for** и заканчиваются служебным словом **end**. Между указанными служебными словами располагаются другие операторы.

Синтаксис оператора условного перехода приведен на рисунке 4.5.

**if условие 1**
**<операторы 1>**
**elseif условие 2**
**< операторы 2>**
**. . .**
**else**
**<операторы>**
**end**

Рисунок 4.5

При выполнении этого оператора сначала проверяется выполнение первого условия. Если оно ложно, то по порядку проверяется

условие в каждом из следующих за ним операторов **ElseIf** (их может быть сколько угодно). Как только находится истинное условие, выполняются соответствующие операторы, после чего выполнение блока **If** прекращается и управление передается на оператор, следующий за словом **End**. Если никакое из проверенных условий **ElseIf** не справедливо, то выполняются операторы, следующие за словом **Else**, если они есть. После чего выполнение блока **If…end** прекращается.

Пример программы с применением условного оператора для вычисления значения ступенчатой функции

$$Y = \begin{cases} 2x, & \text{при} \quad x = 1; \\ x^2, & \text{при} \quad x = 2; \\ x^3, & \text{при } x \neq 1 \text{ и } x \neq 2 \end{cases}$$

и результаты ее выполнения приведены на рисунке 4.6.

```
Command Window
>> x=3;
>> if x==1
y=x*2
elseif x==2
y=x^2
else
y=x^3
end
y =
   27
```

Рисунок 4.6

12 Циклы с неопределенным количеством повторений позволяет задавать конструкция **While…end**, в которой указывается условие продолжения цикла:

**While условие**
**...  [операторы]**
**end**

Цикл повторяется до тех пор, пока условие не станет ложным. После чего все операторы до оператора **end** включительно пропускаются, и программа выполняется дальше.

На рисунке 4.7,*а* приведен пример программы с применением оператора **While**, а на рисунке 4.7,*б* – результаты ее работы.

```
Command Window
>> x=1;
>> while x<=5
y=x^2;
disp([x,y])
x=x+1;
end
```

*а*)

```
Command Window
     1     1
     2     4
     3     9
     4    16
     5    25
fx >>
```

*б*)

Рисунок 4.7

13 В MATLAB имеется команда **disp**, которая осуществляет вывод значений указанной переменной или указанного текста в командное окно. Обращение к ней имеет следующий вид:

**disp (<переменная или текст в апострофах>)**

Чтобы вывести значения нескольких переменных в одну строку (например, при создании таблиц данных), нужно создать единый объект, который содержал бы все эти значения. Это можно сделать, объединив соответствующие переменные в вектор, пользуясь операцией создания вектора строки:

**x = [x1 x2 ... ].**

На рисунке 4.8 приведен пример программы, в результате выполнения которой в командное окно в одну строку выводятся значения четырех ранее определенных переменных через пробелы.

```
Command Window
x1=-3.01; x2=-2.5; x3=-4.3; x4=3.33;
disp([x1 x2 x3 x4])
   -3.0100   -2.5000   -4.3000    3.3300
fx >>
```

Рисунок 4.8

Для одновременного вывода символьной и цифровой информации в командное окно удобно использовать функцию **sprintf**. На рисунке 4.9 приведен пример соответствующей программы.

```
Command Window
>> disp(sprintf('    x1 = %g        x2= %g ',x1,x2));
    x1 = -3.01        x2= -2.5
fx >>
```

Рисунок 4.9

14 В MATLAB для организации цикла с заданным числом повторений используется оператор **For…end**, имеющий следующий формат:

**for Имя = <Начальное значение>:<Шаг>:<Конечное значение>**
**. . . операторы**
**end**

где **Имя** – имя управляющей переменной цикла. Если шаг не указан, то по умолчанию он равен единице.

Конструкция выполняет операторы, следующие за оператором **For**, пока в программе не встретится оператор **end**. Тогда к текущему значению переменной цикла прибавляется значение шага, и полученное значение сравнивается с конечным значением. Если значение переменной цикла больше конечного значения, то выполняется оператор, следующий за оператором **end**, иначе управление снова передается к оператору, находящемуся за оператором **For**.

На рисунке 4.10 приведен пример циклической программы, выводящей на дисплей значения переменной *i*, изменяющиеся от 0 до 30 с шагом, равным 10.

```
Command Window
>> for i=0:10:30;disp (i);end;
    0
    10
    20
    30
fx >>
```

Рисунок 4.10

15 Запишите в среде MATLAB программу для вычисления и вывода на экран в виде таблицы значений $x$ и функции $A(x)$ для заданного варианта из таблицы 1. Для всех вариантов значения $x$ меняются

от –10 до 10 с шагом 5. Для всех вариантов составьте две программы: с использованием оператора **For…end** и оператора **While…end**.

*Отчет*

Включить в отчет по лабораторной работе содержимое окна редактирования и командного окна, полученное при выполнении пп. 11–15.

Таблица 4.1

| Номер варианта | Функция | Номер варианта | Функция |
|---|---|---|---|
| 1 | $A = \sum_{n=1}^{5} \sqrt[4]{1000n + x^4}$ | 11 | $A = \prod_{k=1}^{4} \sin\,(x + 3k)$ |
| 2 | $A = \prod_{n=1}^{8} (x^2 - n)$ | 12 | $A = \sum_{n=1}^{5} \sqrt{n^2 + 2x}$ |
| 3 | $A = \prod_{k=1}^{6} \frac{\cos x}{2k}$ | 13 | $A = \prod_{k=1}^{8} \sqrt{(xk + 2)^3}$ |
| 4 | $A = \sum_{k=1}^{4} (x^4 + 4k)$ | 14 | $A = \sum_{n=1}^{6} \frac{x^n}{n}$ |
| 5 | $A = \prod_{k=1}^{5} (\mathrm{e}^x + kx)$ | 15 | $A = \sum_{k=0}^{5} \mathrm{tg}\,(x + k)$ |
| 6 | $A = \sum_{k=1}^{10} \frac{1-x}{k}$ | 16 | $A = \sum_{i=1}^{10} \frac{1+x}{i+1}$ |
| 7 | $A = \sum_{i=1}^{8} \frac{1+x}{i^3}$ | 17 | $A = \prod_{k=1}^{5} \sqrt{(xk + 1)^2}$ |
| 8 | $A = \prod_{k=1}^{6} \frac{\sin x}{2k + 1}$ | 18 | $A = \sum_{i=1}^{8} \frac{(x)}{i(i+1)}$ |
| 9 | $A = \sum_{i=1}^{7} \frac{x}{(2i+1)^2}$ | 19 | $A = \sum_{i=1}^{6} \frac{(x+2)}{i(i-1)}$ |
| 10 | $A = \sum_{i=1}^{8} \frac{(-1)^x}{(x+1)(i+2)}$ | 20 | $A = \sum_{i=1}^{5} \frac{(x)^2}{4^i + 5^i}$ |

## Контрольные вопросы

1 Как задается место хранения рабочих файлов в MATLAB?

2 Чем отличаются файлы-сценарии и файлы-функции в среде MATLAB?

3 Как создаются, открываются, сохраняются и запускаются на исполнение *M*-файлы в среде MATLAB?

4 Как выполнить несколько строк из окна редактирования в среде MATLAB?

5 Какие операторы управления вычислительным процессом существуют в среде MATLAB и как они работают?