

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ЯДЕРНОЙ ЭНЕРГИИ И ПРОМЫШЛЕННОСТИ

С.А. Абейдулин, С.В. Гайдук, Е.В. Скрыбина

ПРИМЕНЕНИЕ ЭВМ В ИНЖЕНЕРНЫХ РАСЧЕТАХ И МОДЕЛИРОВАНИИ

Учебно-методическое пособие
для выполнения практических
и лабораторных работ по дисциплине

Севастополь
СевГУ
2021

УДК 004.9(075.8)
ББК 32.97я73
А142

Р е ц е н з е н т:

А.А. Скидан - канд. техн. наук, доцент, заведующий кафедрой
«Системы контроля и управления атомных станций»

Абейдулин С.А.

А142 Применение ЭВМ в инженерных расчетах и моделировании: учебно-методическое пособие для выполнения практических и лабораторных работ по дисциплине / Севастопольский государственный университет, Институт ядерной энергии и промышленности, кафедра «Электроэнергетические системы атомных станций» ; С.А. Абейдулин, С.В. Гайдук, Е.В. Скрыбина. – Севастополь: Изд-во СевГУ, 2021. – 47 с.: ил. – Текст : электронный.

Предназначено для обучающихся очной и заочной форм обучения.

Учебно-методическое пособие соответствует программе подготовки бакалавров по направлению: 13.03.02. Электроэнергетика и электротехника.

УДК 004.9(075.8)
ББК 32.97я73

Рассмотрено и рекомендовано на заседании кафедры «Электроэнергетические системы атомных станций» ФГАОУ ВО «Севастопольский государственный университет», протокол № 8/21 от 7 апреля 2021 г.

ОГЛАВЛЕНИЕ

Введение	4
Глава 1. Лабораторные работы.....	5
1.1 Лабораторная работа № 1.....	5
1.2 Лабораторная работа № 2.....	16
1.3 Лабораторная работа № 3.....	22
1.4 Лабораторная работа № 4.....	30
1.5 Лабораторная работа № 5.....	35
Глава 2. Практические работы.....	39
2.1 Практическая работа № 1.....	39
2.2 Практическая работа № 2.....	40
2.3 Практическая работа № 3.....	41
2.4 Практическая работа № 4.....	43
2.5 Практическая работа № 5.....	45
Библиографический список.....	47

ВВЕДЕНИЕ

Цель данного учебного пособия – оказать помощь обучающимся в изучении дисциплины «Применение ЭВМ в инженерных расчетах и моделировании».

Курс относится к числу дисциплин, изучаемых при подготовке бакалавров по направлению подготовки: «Электроэнергетика и электротехника», он предназначен для обучения студентов основам вычисления и моделирования в объеме, необходимом для инженеров электротехнических специальностей.

Общей задачей курса является привитие обучающимся знаний в теории вычислительной математики, выработка навыков в расчёте прикладных задач и умений в моделировании электрических схем и устройств для применения их в вопросах расчета и эксплуатации электрооборудования.

Цель лабораторных занятий – закрепить знания теоретических положений курса, обучить методам применения ЭВМ для расчетов инженерных задач, привить навыки научного анализа и обобщения полученных результатов, а также развитию у обучающихся творческой инициативы и самостоятельности в работе.

Практические занятия направлены на формирование умений и навыков в моделировании электрических схем и устройств с помощью пакета программ Matlab Simulink. В результате изучения курса, обучающиеся научатся не только моделировать электрические устройства и схемы, но и анализировать процессы, происходящие в них.

В пособии содержатся описания лабораторных работ по курсу и задания для расчетов на практических занятиях в каждом из которых указаны тема, учебные цели, программа выполнения работы и рекомендации по оформлению отчета.

Глава 1. Лабораторные работы

1.1 Лабораторная работа № 1

Тема: «Основы работы с системой научных и инженерных расчетов Matlab».

Цель работы:

1. Ознакомление с системой научных и инженерных расчетов Matlab;
2. Получение начальных сведений об окне управления, окне встроенного редактора;
3. Ознакомление с простейшими операциями с числами, векторами и матрицами, элементарными математическими функциями;
4. Создание М-файлов.

Постановка задачи: используя теоретические сведения, выполнить задания лабораторной работы и подготовить отчет.

Краткие теоретические сведения о системе Matlab

Система MatLAB (сокращение от MATrix LABoratory — МАТричная ЛАБоратория) является интерактивной системой для выполнения инженерных и научных расчетов, ориентированной на работу с массивами данных. MATLAB представляет собой интерактивную среду для вычислений и моделирования.

- Может работать как в *режиме непосредственных вычислений* (напоминает режим «командной строки» в ОС DOS), так и в *режиме интерпретации написанных программ*.
- Сильная сторона системы – *работа с матрицами и векторами*. Численное значение или аналитическая формула, а также сообщения системы выводятся на экран в виде списка.
- Помимо обычных алгебраических вычислений система имеет *обширный набор встроенных функций*, а также имеется *возможность создавать пользовательские функции*.
- В системе очень качественно реализовано *построение двух и трехмерных изображений, в том числе динамически изменяющихся*.
- Кроме того, имеется библиотека, которая обеспечивает удобное управление исполнением программ.

И это только базовый набор, который обычно расширяется многочисленными дополнениями – например языком моделирования нелинейных динамических систем – *Simulink*. Основное назначение пакета — это технические расчеты. Система использует математический сопроцессор и допускает возможность обращения к программам, написанным на языках FORTRAN, C и C++.

Возможности системы

В области математических вычислений:

- матричные, векторные, логические операторы;
- элементарные и специальные функции;
- полиномиальная арифметика;

- многомерные массивы;
- массивы записей, массивы ячеек.

В области реализации численных методов:

- дифференциальные уравнения;
- вычисление одномерных и двумерных квадратур;
- поиск корней нелинейных алгебраических уравнений;
- оптимизация функций нескольких переменных;
- одномерная и многомерная интерполяция.

В области программирования:

- свыше 500 встроенных математических функций;
- ввод/вывод двоичных и текстовых файлов;
- применение программ, написанных на языках высокого уровня;
- автоматическая перекодировка процедур MATLAB в тексты программ на языках Си и C++;

- типовые управляющие структуры.

В области визуализации и графики:

- возможность создания двумерных и трехмерных графиков;
- осуществление визуального анализа данных и т.д.

В MATLAB реализована удобная операционная среда, позволяющая формулировать проблемы и получать решения в привычной математической форме, не прибегая к рутинному программированию.

Ориентация на матричные операции

Система MATLAB выполняет сложные и трудоемкие операции над векторами и матрицами даже в режиме прямых вычислений. Наряду с обычными арифметическими и алгебраическими действиями могут использоваться такие сложные операции, как инвертирование матрицы, вычисление ее собственных значений, решение систем линейных уравнений и т.п. Обычные числа и переменные в MATLAB рассматриваются как матрицы размера 1×1 , что дает единообразные формы и методы проведения операций над обычными числами и массивами.

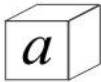
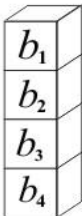
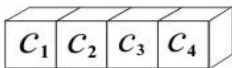
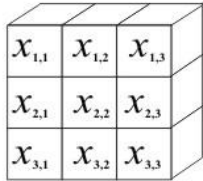
число	вектор-столбец	вектор-строка	матрица (квадратная)
			
<pre>»a=2.34 ans=2.34</pre>	<pre>»b=[3;4;8;12] ans= 3 4 8 12</pre>	<pre>»b=[3 4 8 12] ans= 3 4 8 12</pre>	<pre>»b=[2,3,5;4,8,2;12,-3,7] ans= 2 3 5 4 8 2 12 -3 7</pre>

Рисунок 1.1.1

Это обеспечивает и упрощение записи операций, производимых одновременно над всеми элементами векторов и матриц и что большинство функций может работать с аргументами в виде векторов и матриц. При необходимости вектора и матрицы преобразуются в массивы, и значения вычисляются для каждого их элемента.

MATLAB - расширяемая система, поэтому ее легко приспособить к решению самых разных классов задач. Одна из отличительных черт системы — это легкость ее модификации и адаптации к конкретным задачам пользователя. Пользователь может ввести в систему любую новую команду, оператор или функцию и пользоваться затем ими так же просто, как и встроенными операторами и функциями. При этом нет необходимости в их предварительном описании.

Новые программы, функции и процедуры в системе *MatLAB* сохраняются в виде файлов, имеющих расширение *.m, то есть реализуется в виде, так называемых, *m-файлов*. Расширения системы хранятся на жестком диске компьютера и в нужный момент вызываются для использования точно так же, как встроенные в *MATLAB* (внутренние) функции и процедуры. Благодаря текстовому формату *m-файлов* пользователь может ввести в систему любую новую команду, оператор или функцию и затем пользоваться ими также, как и встроенными операторами или функциями. *MATLAB* содержит огромное число операторов и функций, которые позволяют решать множество практических задач. К примеру, это функции обращения или транспонирования матриц, вычисления значений производной или интеграла и другие функции. Число таких функций с учетом пакетов расширения системы уже достигает многих тысяч и непрерывно увеличивается.

MatLAB имеет большие возможности для работы с сигналами, для расчета и проектирования аналоговых и цифровых фильтров, для построения их частотных, импульсных и переходных характеристик. Имеются в наличии и средства для спектрального анализа и синтеза, в частности, для реализации прямого и обратного преобразования Фурье. Благодаря этому система довольно привлекательна для проектирования электронных устройств.

MATLAB с момента своего создания разрабатывался как мощный математико-ориентированный язык программирования высокого уровня. Запись программ в системе традиционна и потому привычна для большинства пользователей, имеющих навыки написания программ. К тому же система дает возможность редактировать программы с помощью любого привычного для пользователя текстового редактора. Имеется и собственный редактор с отладчиком.

Работа в среде *Matlab* может осуществляться в двух режимах:

- в режиме калькулятора, когда вычисления производятся непосредственно после набора очередного оператора или команды *Matlab*; при этом значения результатов вычисления могут присваиваться некоторым переменным, либо результаты получаются непосредственно, без присваивания (как в обычных калькуляторах);

• путем вызова программы, составленной и записанной на диске на языке Matlab, которая содержит все необходимые команды, обеспечивающие ввод данных, организацию вычислений и вывод результатов на экран (программный режим).

В обоих режимах пользователю доступны практически все вычислительные возможности системы, в том числе по выводу информации *в графической форме*. Программный режим позволяет сохранять разработанные вычислительные алгоритмы и, таким образом, повторять вычисления при других исходных данных.

Основные преимущества системы Matlab:

- удобство пользовательского интерфейса,
- высокие вычислительные возможности (богатая библиотека)
- широкая область применения результатов расчета,
- открытость и постоянная наращиваемость архитектуры.

Командное окно системы MATLAB

После вызова MATLAB из среды Windows на экране появляется командное окно среды MATLAB (рис. 1.1.2).

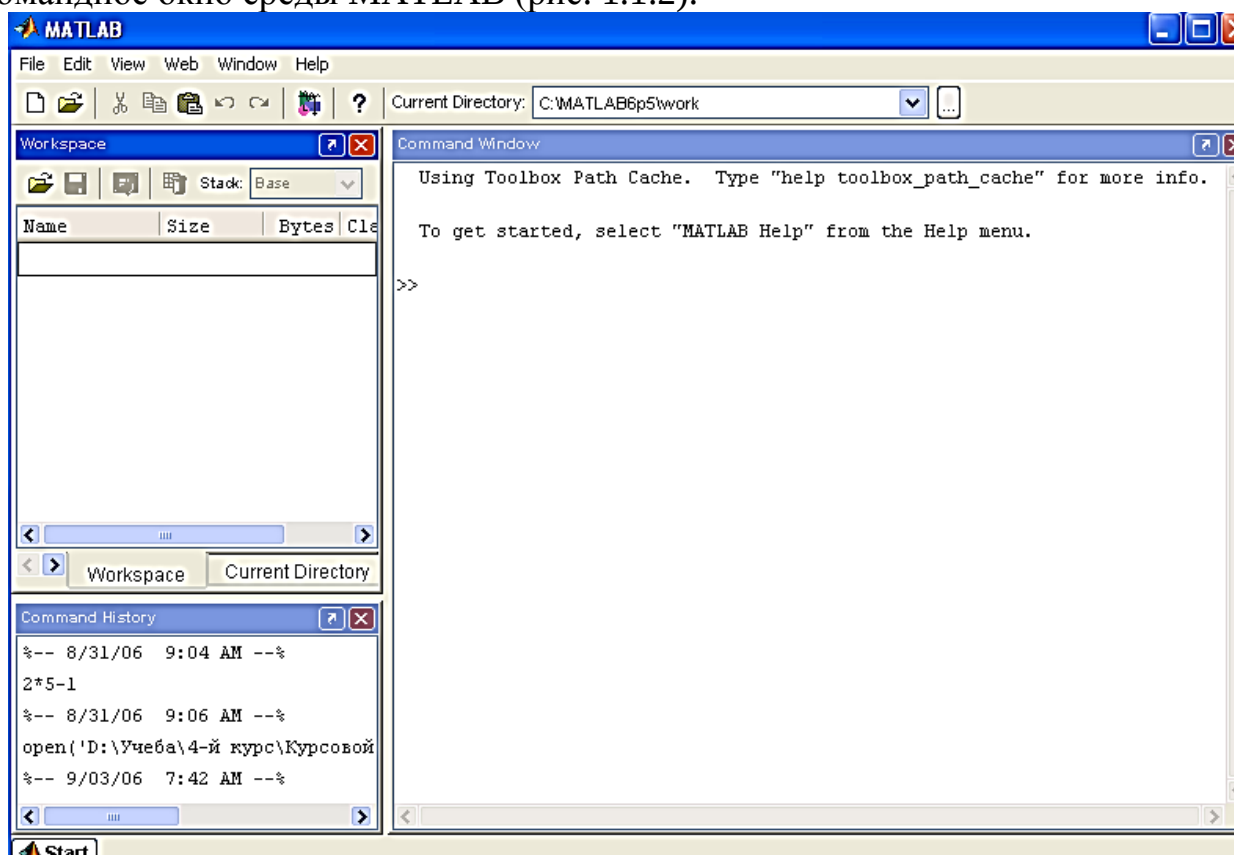


Рисунок 1.1.2 – Командное окно системы MATLAB

Это окно является основным в системе MATLAB. В нем отображаются символы команд, которые набираются пользователем, результаты выполнения этих команд, текст исполняемой программы, а также информация об ошибках выполнения программы, распознанных системой. Признаком того, что система готова к восприятию и выполнению очередной команды, является наличие в

последней строке текстового поля окна знака приглашения (`>>`), после которого возможен ввод символов команд и другие операции.

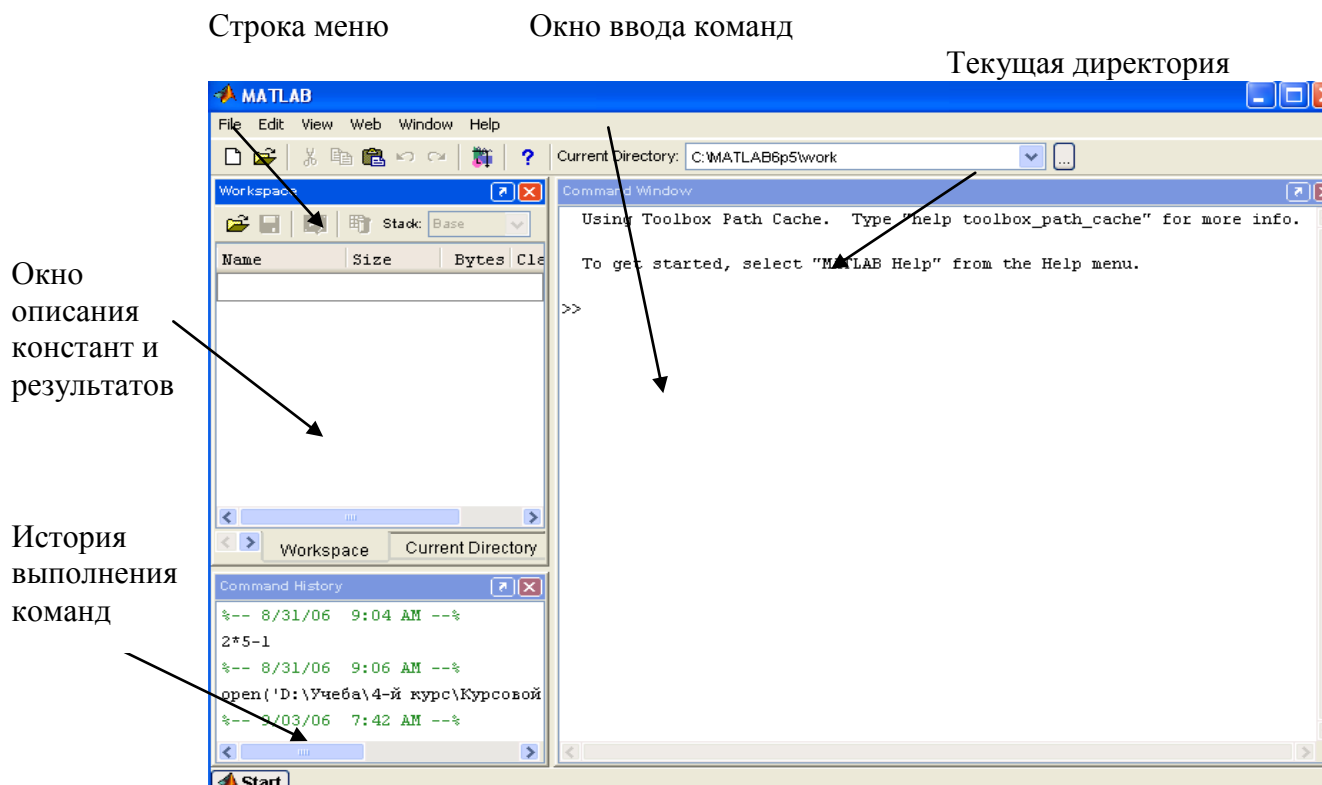


Рисунок 1.1.3 – Интерфейс системы MATLAB

Основы работы в среде MATLAB

Ввод действительных чисел.

Ввод чисел с клавиатуры производится по общим правилам, принятым для языков программирования высокого уровня:

- для отделения дробной части числа применяется десятичная точка (вместо запятой при обычной записи);
- десятичный показатель числа записывается в виде целого числа после предварительной записи символа *e* (пример: $1.2305e-5$).

Простейшие арифметические операции:

+ сложение; - вычитание; * умножение; / деление слева направо;
\ деление справа налево; ^ возведение в степень.

Использование MATLAB в режиме калькулятора может происходить путем простой записи в командную строку последовательности арифметических действий с числами, например:

```
>> 4.5^2*7.23-3.14*10.4  
ans = 113.7515
```

Результат выполнения действий выводится как значение системной переменной **ans**. Вывод промежуточной информации в командное окно подчиняется следующим правилам:

- если запись оператора **не заканчивается** символом ";", результат действия этого оператора сразу же выводится в командное окно;

- если оператор заканчивается символом ";", результат его действия не отображается в командном окне;

- если оператор не содержит знака присваивания (=), т.е. является просто записью некоторой последовательности действий над числами и переменными, то значение результата присваивается специальной системной переменной **ans**;

- полученное значение можно использовать в последующих операторах вычислений под именем **ans**; при этом следует помнить, что значение системной переменной **ans** изменяется после действия очередного оператора без знака присваивания;

- в общем случае форма вывода результата в командное окно имеет вид:

<имя переменной> = <результат>

Особенностью системы MATLAB как калькулятора является возможность использования имен переменных для записи промежуточных результатов в память. Для этого применяется операция присваивания в соответствии со схемой:

<имя переменной> = <выражение>;
(например, $x=25+17$)

В системе MATLAB имеется несколько **зарезервированных имен переменных**:

i и j – мнимая единица (корень квадратный из -1);

π – число π (3,1416);

$\inf$ – обозначение машинной бесконечности;

NaN – обозначение неопределенного результата (например типа 0/0 или $\inf/\inf$).

Ввод комплексных чисел.

Ввод комплексного числа производится путем записи в командном окне строки следующего типа:

<имя переменной> = <значение ДЧ> + i (j) * <значение МЧ> ,

где ДЧ - действительная часть комплексной величины;

МЧ – мнимая часть.

Например: $x=3+j*2$

Простейшие арифметические операции с комплексными числами:

+ сложение; - вычитание; * умножение; / деление слева направо;

\ деление справа налево; ^ возведение в степень.

В MATLAB есть несколько **дополнительных функций, рассчитанных только на комплексный аргумент**:

- $\text{real}(Z)$ - выделяет действительную часть комплексного аргумента Z ;

- $\text{imag}(Z)$ - выделяет комплексную часть комплексного аргумента Z ;

- $\text{angle}(Z)$ - вычисляет значение аргумента комплексного числа Z (в радианах от $-\pi$ до $+\pi$);

- $\text{conj}(Z)$ - выдает число, комплексно сопряженное относительно Z .

Элементарные математические функции

Общая форма вызова функций MATLAB имеет следующий вид:

<имя результата> = <имя функции> (<список имен аргументов или их значений>)

В языке MATLAB предусмотрены следующие элементарные математические функции:

Тригонометрические и гиперболические функции

sin(Z)	синус числа Z
sinh(Z)	гиперболический синус
asin(Z)	арксинус (в радианах, в диапазоне от $-\pi/2$ до $+\pi/2$)
asinh(Z)	обратный гиперболический синус
cos(Z)	косинус
cosh(Z)	гиперболический косинус
acos(Z)	арккосинус (в диапазоне от 0 до π)
acosh(Z)	обратный гиперболический косинус
tan(Z)	тангенс
tanh(Z)	гиперболический тангенс
atan(Z)	арктангенс (в диапазоне от $-\pi/2$ до $+\pi/2$)
atan2(X,Y)	четырёхквadrантный арктангенс (угол в диапазоне от $-\pi$ до $+\pi$ между горизонтальным правым лучом и лучом, проходящим через точку с координатами X и Y)
atanh(Z)	обратный гиперболический тангенс
sec(Z)	секанс
sech(Z)	гиперболический секанс
asec(Z)	арксеканс
asech(Z)	обратный гиперболический секанс
csc(Z)	косеканс
csch(Z)	гиперболический косеканс
acsc(Z)	арккосеканс
acsch(Z)	обратный гиперболический косеканс
cot(Z)	котангенс
coth(Z)	гиперболический котангенс
acot(Z)	арккотангенс
acoth(Z)	обратный гиперболический котангенс

Экспоненциальные функции

exp(Z)	экспонента числа Z
log(Z)	натуральный логарифм
log10(Z)	десятичный логарифм
sqrt(Z)	квадратный корень из числа Z
abs(Z)	модуль числа Z
round(Z)	обычное округление числа Z до ближайшего целого

Простейшие операции с векторами и матрицами

MATLAB является системой, которая специально предназначена для осуществления сложных вычислений с векторами, матрицами и полиномами.

Под вектором в MATLAB понимается одномерный массив чисел, а под матрицей - двумерный массив.

Исходные значения векторов-строк можно задавать путем поэлементного ввода. Для этого вначале указывают *имя вектора*, затем ставят знак присваивания (=), далее открывающую квадратную скобку ([), за ней значения вектора, отделяя их между собой *пробелами* или *запятыми*. Завершается запись закрывающей квадратной скобкой (]).

Например: $V=[1\ 2\ 3]$ или $V=[1,2,3]$

$V = \quad 1 \quad 2 \quad 3$

Длинный вектор можно вводить частями, которые затем объединяют с помощью операции объединения векторов в строку.

$V1 = [1\ 2\ 3]; \quad V2 = [4\ 5\ 6];$

$V = [V1\ V2]$

$V = 1\ 2\ 3\ 4\ 5\ 6$

Язык MATLAB дает пользователям возможность сокращенного ввода вектора, элементы которого являются арифметической прогрессией.

$V = nz : h : kz,$

где nz – начальное значение прогрессии (первый элемент вектора);

kz – конечное значение прогрессии (последний элемент вектора);

h – разность прогрессии (шаг).

Например: $V = -0.1:0.3:1.4$

$V = -0.1\ 0.2\ 0.5\ 0.8\ 1.1\ 1.4$

Вектор-столбец задается аналогично вектору строке, но элементы отделяются друг от друга знаком «;».

Ввод элементов матрицы осуществляется по строкам. При этом элементы строки матрицы отделяются друг от друга пробелами или запятыми, а строки отделяются друг от друга знаком «;».

Например: $A = [1,2,3; 4,5,6; 7,8,9]$

$A = \quad 1\ 2\ 3$

$\quad 4\ 5\ 6$

$\quad 7\ 8\ 9$

Формирование векторов и матриц

MATLAB имеет несколько функций, которые позволяют формировать векторы и матрицы определенного вида:

$\text{zeros}(M, N)$ – создает матрицу размером M на N с нулевыми элементами;

$\text{zeros}(2,3)$

$ans = \quad 0 \quad 0 \quad 0$

$\quad 0 \quad 0 \quad 0$

$\text{ones}(M, N)$ - создает матрицу размером M на N с единичными элементами;

$\text{ones}(2,3)$

$ans = \quad 1 \quad 1 \quad 1$

$\quad 1 \quad 1 \quad 1$

`eye (M, N)` - создает матрицу размером M на N с единицами по главной диагонали и всеми остальными нулями;

`eye(2,3)`

`ans =` 1 0 0
 0 1 0

`rand (M,N)` - создает матрицу размером M на N из случайных чисел, равномерно распределенных в диапазоне от 0 до 1;

`rand(2,3)`

`ans =` 0.9501 0.6068 0.8913
 0.2311 0.4860 0.7621

Извлечение и вставка отдельных элементов матриц

Обращение к любому элементу матрицы осуществляется указанием после имени матрицы номера строки и номера столбца на пересечении которых расположен элемент матрицы.

Например, $A =$ 1 2 3

 4 5 6

 7 8 9

$X = A(2,3)$

$X = 6$

Если нужно наоборот вставить на это место какое-либо число, то это можно сделать следующим образом:

$A(2,3) = \pi$

$A =$ 1 2 3
 4 5 3.14
 7 8 9

Пусть требуется создать вектор $V1$, состоящий из элементов третьего столбца матрицы A .

$V1 = A(:,3)$

$V1 =$ 3
 3.14
 9

Пусть требуется создать матрицу B размером 2:2 состоящую из элементов правого верхнего угла матрицы A .

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

$B =$ 2 3
 5 3.14

Пусть требуется внести элементы матрицы B в левый нижний угол матрицы A .

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

$A =$ 1 2 3
 2 3 6
 5 3.14 9

Растянуть матрицу в один вектор можно с помощью следующей записи:

```
A= 1 2 3
    4 5 6
V=A(:)
V=1 2 3 4 5 6
```

Действия над векторами

Выделим 2 группы действий над векторами: *векторные* действия и действия *поэлементному* преобразованию векторов.

Векторные действия:

- 1) Сложение векторов: $x=[1\ 2\ 3]$; $y=[4\ 5\ 6]$; $v=x+y$ ($v=5\ 7\ 9$)
- 2) Вычитание векторов: $v=x-y$ ($v=-3\ -3\ -3$)
- 3) Транспонирование вектора: x' $ans = 1$
2
3
- 4) Умножение вектора на число: $v = x*2$ ($v=2\ 4\ 6$)
- 5) Умножение 2 векторов (для векторов одинаковой длины, один строка, а другой столбец):
 $x=[1\ 2\ 3]$; $y=[4\ 5\ 6]$; $v=x'*y$ $v = \begin{matrix} 4 & 5 & 6 \\ 8 & 10 & 12 \\ 12 & 15 & 18 \end{matrix}$
 $v=x*y'$ $v = 32$

Действия поэлементному преобразованию векторов:

Все эти операции преобразуют элементы вектора как элементы обычного одномерного массива чисел. К таким операциям относятся все из вышеперечисленных **элементарных математических функций**, зависящих от одного аргумента.

Например: $x=[-2,-1,0,1,2]$

$V=\sin(x)$; $V=-0.9093\ -0.8415\ 0\ 0.8415\ 0.9093$

$V=\tan(x)$; $V=\exp(x)$ и др.

- 1) Добавление (вычитание) числа к каждому из элементов (+, -);
- 2) Поэлементное умножение векторов (.*);
- 3) Поэлементное деление векторов (./, ./)
- 4) Поэлементное возведение в степень.(.^)

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ № 1

Тема: «Основы работы с системой научных и инженерных расчетов Matlab»

1. Осуществить ввод действительного числа: $m, nk \cdot 10^{-7}$.
2. Выполнить простую арифметическую операцию $m, n/k \cdot t, n - 0,001^2 \cdot 3,14$
3. Осуществить ввод комплексного числа, действительная часть которого равна k , а мнимая равна $-n$.
4. Выполнить простую арифметическую операцию с двумя комплексными числами.
5. Вычислить значение одной из элементарных математических функций (смотри стр. 7 и 8).
6. Сформировать вектор из 5 любых неотрицательных элементов.
7. Сформировать матрицу размером 3×4 с 1 по главной диагонали и нулевыми остальными элементами.
8. В созданной матрице извлечь элемент 2-й строки и 3-столбца.
9. Создать 2 вектора x и y по 3 элемента каждый и провести операции сложения, вычитания, транспонирования векторов, и их перемножения.

m - месяц рождения

n - дата (десятки) рождения

k - дата (единицы) рождения

t - последняя цифра года рождения.

Оформить отчёт к лабораторной работе в виде файла и документа.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Содержание каждого задания.
4. Текст реализации в MATLAB с комментариями (комментарии отделяются знаком %).
5. Выводы по работе.

1.2 Лабораторная работа № 2

Тема: «Работа с массивами в Matlab»

Цель работы:

1. Ознакомление с системой научных и инженерных расчетов Matlab;
2. Научиться выполнять вычисления с векторами и массивами в MATLAB.

Постановка задачи: используя теоретические сведения, выполнить задания лабораторной работы и подготовить отчет.

Краткие теоретические сведения

Работа с векторами.

Представление данных в виде матриц используется при обработке какого-то массива экспериментальных данных, полученных в ходе эксперимента, также при исследовании и анализе различных преобразовательных процессов для представления вектора входных воздействий и получении вектора выходных величин.

Элементы вектора-столбца отделяются точкой с запятой, а элементы строки отделяются пробелом или запятой:

```
>> a=[5 6 7]
```

```
a =
```

```
5 6 7
```

```
>> b=[1; 2; 3]
```

```
b =
```

```
1
```

```
2
```

```
3
```

Транспонирование векторов осуществляется с помощью апострофа.

```
>> a'
```

```
a =
```

```
5
```

```
6
```

```
7
```

Для определения длины векторов используется функция `length`. Для определения размерности массивов используются функции `ndims` и `size`.

В MATLAB операции сложения, вычитания и элементарные функции выполняются с массивами поэлементно.

```
>> sin(a)
```

```
ans =
```

```
-9.5892e-001
```

```
-2.7942e-001
```

```
6.5699e-001
```

```
>> a+b
```

```
ans =
```

```
6
```

```
8
```

При умножении векторов a и b без точки перед знаком умножения MATLAB выдаст сообщение об ошибке, так как при произведении матрицы на матрицу необходимо согласование размерностей.

```
>> a*b
```

```
??? Error using ==> mtimes
```

```
Inner matrix dimensions must agree.
```

При поэлементных операциях необходимо ставить точку перед знаками $*$, $^$, $/$, $\backslash$.

```
>> a.*b
```

```
ans =
```

```
5
```

```
12
```

```
21
```

Перемножение элементов вектора осуществляется функцией `prod`, суммирование элементов вектора – функцией `sum`. Минимум и максимум находятся с помощью функций `max` и `min`. Упорядочение вектора по возрастанию элементов осуществляется функцией `sort(a)`, по убыванию – `sort(-a)`.

Для вычисления скалярного произведения нужно использовать функцию `sum` и поэлементное умножение. Для векторного произведения применяется функция `cross`.

Двумерные массивы, матрицы.

При задании матриц элементы в строках вводятся через пробел или запятую, а строки друг от друга отделяются точкой с запятой.

Пример. $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

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

```
A =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

Обращение к элементам матрицы осуществляется с помощью двух индексов – номеров строки и столбца в круглых скобках.

```
>> A(2,3)
```

```
ans =
```

```
6
```

При сложении или вычитании матрицы должны быть одного размера, иначе MATLAB выдаст сообщение об ошибке. Транспонирование матрицы производится так же, как транспонирование вектора, при помощи символа $'$.

Для решения систем линейных уравнений используется символ $\backslash$.

$$\begin{bmatrix} 1.2x_1 + 0.3x_2 - 0.2x_3 = 1.3 \end{bmatrix}$$

Пример.
$$\begin{cases} 0.5x_1 + 2.1x_2 + 1.3x_3 = 3.9 \\ -0.9x_1 + 0.7x_2 + 5.6x_3 = 5.4 \end{cases}$$

При решении указанной системы необходимо ввести матрицу коэффициентов перед x - A , матрицу правой части - B .

```
>> A=[1.2 0.3 -0.2; 0.5 2.1 1.3; -0.9 0.7 5.6]
```

```
A =
```

```
    1.2000    0.3000   -0.2000
    0.5000    2.1000    1.3000
   -0.9000    0.7000    5.6000
```

```
>> b=[1.3; 3.9; 5.4]
```

```
b =
```

```
    1.3000
    3.9000
    5.4000
```

```
>> x=A\b
```

```
x =
```

```
    1.000
    1.000
    1.000
```

Проверка $b=A*x$.

```
>> A*x
```

```
ans =
```

```
    1.3000
    3.9000
    5.4000
```

MATLAB позволяет конструировать блочные матрицы, но при этом соответствующие размеры блоков должны совпадать. Для выделения блоков матрицы осуществляется индексация при помощи двоеточия.

```
>> A=[1.2 0.3 -0.2; 0.5 2.1 1.3; -0.9 0.7 5.6]
```

```
A =
```

```
    1.2000    3.0000   -2.0000
    5.2000    1.0000    1.3000
   -9.0000    7.0000    5.6000
```

```
>> A1=A(2:3, 2:3)
```

```
A1 =
```

```
    2.1000    1.3000
    7.0000    5.6000
```

Для удаления строки или столбца необходимо использовать пустые парные квадратные скобки []. Удалим первый столбец матрицы A .

```
>> A(:,1) = []
```

```
A =
```

```
    3.0000   -2.0000
```

```
2.1000  1.3000
7.0000  5.6000
```

С помощью индексации можно заполнять матрицы нужными элементами.

```
>> A(:,1)=1
```

A =

```
1.0000 -2.0000
1.0000  1.3000
1.0000  5.6000
```

В MATLAB имеется большая коллекция матриц специального вида. Матрица, состоящая из нулей, вызывается функцией `zeros`, в которой задается количество строк и столбцов.

```
>> Z=zeros (4,3)
```

Z =

```
0  0  0
0  0  0
0  0  0
0  0  0
```

Если нужна квадратная матрица, то достаточно указывать один аргумент.

```
>> Z=zeros (3)
```

Z =

```
0  0  0
0  0  0
0  0  0
```

Единичная матрица содержит единицы на главной диагонали и вызывается командой `eye`.

```
>> E=eye (3,6)
```

E =

```
1  0  0  0  0  0
0  1  0  0  0  0
0  0  1  0  0  0
```

Матрица из единиц задается функцией `ones`.

```
>> T=ones (3,6)
```

T =

```
1  1  1  1  1  1
1  1  1  1  1  1
1  1  1  1  1  1
```

Матрица из чисел, расположенных случайным образом между 0 и 1, задается функцией `rand`.

```
>> R=rand (3)
```

R =

```
0.9649  0.9572  0.1419
0.1576  0.4854  0.4218
0.9706  0.8003  0.9157
```

Матрица из случайных нормально распределенных чисел задается функцией `randn`.

```
>> R=randn (3)
```

R =

```
-0.4326  0.2877  1.1892
-1.6656 -1.1465 -0.0376
 0.1253  1.1909  0.3273
```

Диагональная матрица с необходимыми элементами по диагонали и остальными нулевыми элементами формируется функцией `diag`.

```
>> d= [1 2 3 4];
```

```
>> D=diag(d)
```

D =

```
1  0  0  0
0  2  0  0
0  0  3  0
0  0  0  4
```

Для заполнения не главной, а побочной диагонали используется второй аргумент в функции `diag(d,2)`.

Магическая матрица (магический квадрат), у которой суммы элементов в строках и столбцах равны, вызывается функцией `magic`.

```
>> M=magic (3)
```

M =

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

В MATLAB заложены функции по матричному анализу.

- Функцией `det` можно вычислить определитель матрицы.
- Функция `rank` вычисляет ранг матрицы.
- Функция `norm` выдает векторные и матричные формы.

Как их правильно использовать и какие задавать аргументы, можно узнать из справки.

```
>> help det
```

DET Determinant.

DET(X) is the determinant of the square matrix X.

Use COND instead of DET to test for matrix singularity.

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ № 2

Тема: «Работа с массивами в Matlab»

Номер варианта - последняя цифра в зачётной книжке.

Задание 1. Задать матрицу A с помощью операции конкатенации:

$$A1 = \begin{bmatrix} 3,25 & 1,07 & 2,34 \end{bmatrix}$$

$$A2 = \begin{bmatrix} 10,10 & 0,25 & 4,78 \end{bmatrix}$$

$$A3 = \begin{bmatrix} 5,04 & 7,79 & 3,31 \end{bmatrix}$$

Для четных номеров варианта A1, A2, A3 – строки матрицы, для нечетных – столбцы.

Задание 2. Сгенерируйте массив B размером 3×3 со случайными элементами, равномерно распределенными на интервале от 0 до 10.

Задание 3. Выполните действия:

1) $A + 10 \cdot B$

2) $A * B$

3) почленно умножить A на B,

4) расположить элементы матрицы A по возрастанию (по столбцам),

5) определить максимальный и минимальный элементы матрицы B,

6) вычислить определитель матрицы B,

7) $A \cdot B - B \cdot A$

8) создать случайную матрицу 4×4 M и вычислить

$$(2 \cdot M^3 - 3 \cdot (M^{-1})^2 + E) \cdot M^T,$$

где E – единичная матрица.

Задание 4. Задать массив C из трех элементов.

Задание 5. Решить систему алгебраических линейных уравнений и осуществить проверку.

$$A \cdot X = C.$$

Задание 6. Определить массив D:

$$D = |\sin(A) + B^{3/5}|.$$

Оформить отчёт к лабораторной работе в виде файла и документа.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Содержание каждого задания.
4. Текст реализации в MATLAB с комментариями (комментарии отделяются знаком %).
5. Выводы по работе.

1.3 Лабораторная работа № 3

Тема: «Графические построения в среде MATLAB»

Цель работы:

1. Ознакомление с системой научных и инженерных расчетов Matlab;
2. Ознакомление с простейшими графическими средствами MATLAB.

Постановка задачи: используя теоретические сведения, выполнить задания лабораторной работы и подготовить отчет.

Краткие теоретические сведения

Двухмерная графика.

Для построения в системе MATLAB графика функции одной переменной $y(x)$ в декартовой (прямоугольной) системе координат используется команда

plot (X1, Y1, S1, X2, Y2, S2, ...)

где X_i , Y_i – заданные векторы одинакового размера, элементы которых являются координатами точек i -й кривой; S_i – необязательные символьные переменные, относящиеся к i -й кривой. Любая из S_i может содержать до трех специальных символов, определяющих тип линии, соединяющей отдельные точки графика (табл. 1.3.1), тип точки графика (табл. 1.3.2) и цвет линии (табл. 1.3.3).

Таблица 1.3.1

Обозначение	Тип линии
–	Сплошная
– .	Штрих-пунктир
:	Двойной пунктир
--	Штриховая

Таблица 1.3.2

Обозначение	Тип точки
.	Точка
V	Треугольник (вниз)
o	Окружность
A	Треугольник (вверх)
X	Крест
<	Треугольник (влево)
+	Плюс
>	Треугольник (вправо)
*	Звездочка
P	Пятиугольник
S	Квадрат
H	Шестиугольник
D	Ромб

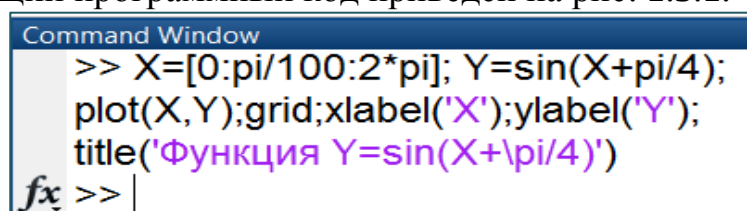
Таблица 1.3.3

Обозначение	Цвет линии
Y	Желтый
G	Зеленый
M	Фиолетовый
B	Синий
C	Голубой
W	Белый
R	Красный
K	Черный

В случае, если значения переменных **S** не указано, то, по умолчанию, принимаются следующие параметры графика: тип линии графика – сплошная; тип точки графика – пиксель; цвет каждого последующего графика устанавливается в следующем порядке: синий, зеленый, красный, голубой, фиолетовый, желтый, черный.

Рассмотрим пример построения функции $Y = 10 \sin(X + \pi/4)$ для значений аргумента, изменяющихся от нуля до 2π с шагом, равным $\pi/100$.

Соответствующий программный код приведен на рис. 1.3.1.



```

Command Window
>> X=[0:pi/100:2*pi]; Y=sin(X+pi/4);
plot(X,Y);grid;xlabel('X');ylabel('Y');
title('Функция Y=sin(X+\pi/4)')
fx >> |

```

Рисунок 1.3.1

Сначала сформируем вектор значений аргумента **X** (при этом греческую букву «π» зададим как «pi»), затем вычислим вектор соответствующих значений функции **Y** и, наконец, построим график зависимости $Y(X)$ с помощью функции **plot**.

Функция **grid** позволяет добавить на график сетку из координатных линий, названия осей задаются с помощью функции **xlabel** и **ylabel**, а заголовок графика выводится с помощью процедуры **title**. Обратите внимание на то, что для вывода в заголовке графика греческой буквы «π» перед символами «pi» добавлена обратная косая черта «\». Аналогично можно вывести и другие прописные и строчные греческие буквы \alpha, \beta, \gamma, \Gamma, \Sigma, \Pi и т.д.

В результате на экране появится окно, в котором будет построен заданный график (рис. 1.3.2).

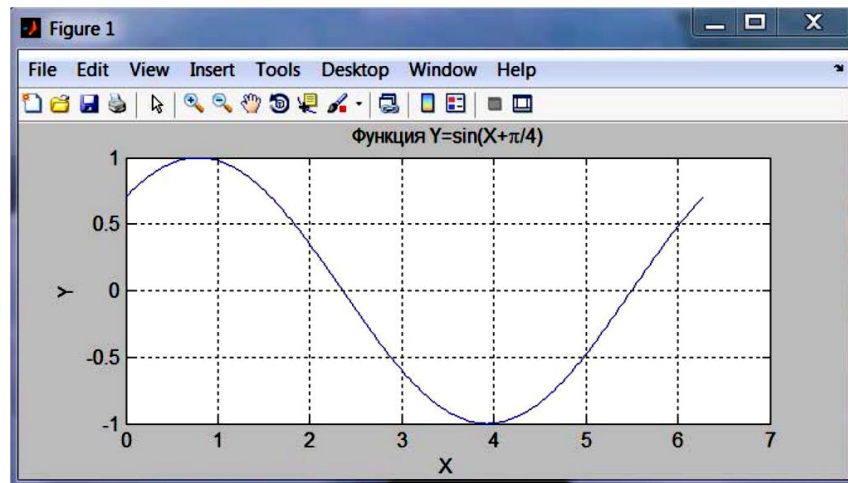


Рисунок 1.3.2

Для получения на графике гладких кривых необходимо выбрать достаточно большое количество точек на графике при формировании вектора аргумента. Если при обращении к функции **plot** вектор аргумента не указан явно, то MATLAB, по умолчанию, в качестве аргумента принимает номера элементов вектора функции (рис. 1.3.3).

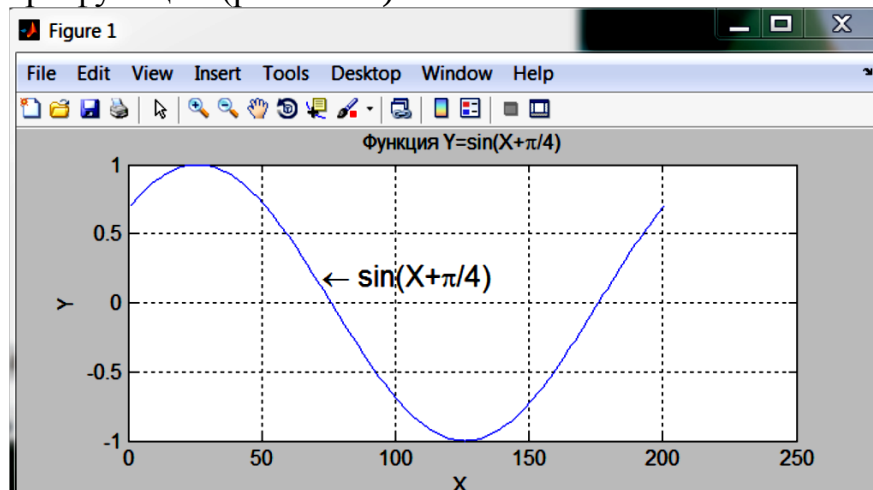


Рисунок 1.3.3

Для добавления текста в определенное место графика, например для обозначения кривой графика, используется команда **text(x, y, 'string')** где **x**, **y** определяют координаты точки начала текста; **'string'** – строковая константа, определяющая вводимый в график текст.

На рисунке 1.3.3 приведен пример текста, вставленного на графике в точку с координатами 70; 0,2. Соответствующий программный код приведен на рис. 1.3.4.

```
Command Window
>> plot(Y);grid;xlabel('X');ylabel('Y');
title('Функция Y=sin(X+\pi/4)')
text(70,0.2,'← sin(X+\pi/4)','FontSize',14)
fx >> |
```

Рисунок 1.3.4

Обратите внимание на то, что строковая константа начинается с обратной косой черты «\» и команды **leftarrow**, которая задает стрелку, направленную влево. Аналогично можно добавить и другие виды стрелок (команды **\uparrow**, **\rightarrow** и т.д.), а также знаки «больше или равно», «меньше или равно», «не равно» (команды **\geq**, **\leq**, **\neq**).

В программе рисунок 1.3.4 для задания требуемого размера шрифта в пунктах используется строковая константа 'FontSize'. В MATLAB имеется возможность задавать и требуемый тип с помощью команд **\fontname**{имя шрифта}, а также начертание шрифта с помощью следующих команд:

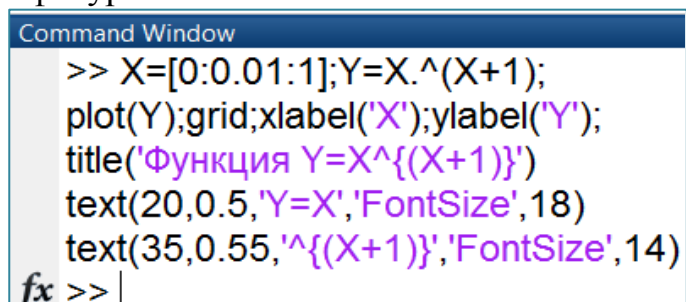
\bf – для жирного текста;

\it – для курсива;

\rm – для обычного текста.

MATLAB позволяет выводить с помощью команды **text** достаточно сложные математические формулы, включающие в себя индексы, дроби, корни, интегралы, суммы и т.д. Подробнее информацию о функции **text** можно посмотреть с помощью команды **doc text**.

Для задания индексов используется символ «^» перед верхним индексом или символ «_» перед нижним индексом. Так как в программе рис. 1.3.5 верхний индекс содержит больше одного символа, то он должен быть полностью заключен в фигурные скобки.



```

>> X=[0:0.01:1];Y=X.^(X+1);
plot(Y);grid;xlabel('X');ylabel('Y');
title('Функция Y=X^{(X+1)}')
text(20,0.5,'Y=X','FontSize',18)
text(35,0.55,'^{(X+1)}','FontSize',14)
fx >> |
  
```

Рисунок 1.3.5

Результаты выполнения данного программного кода приведены на рис. 1.3.6.

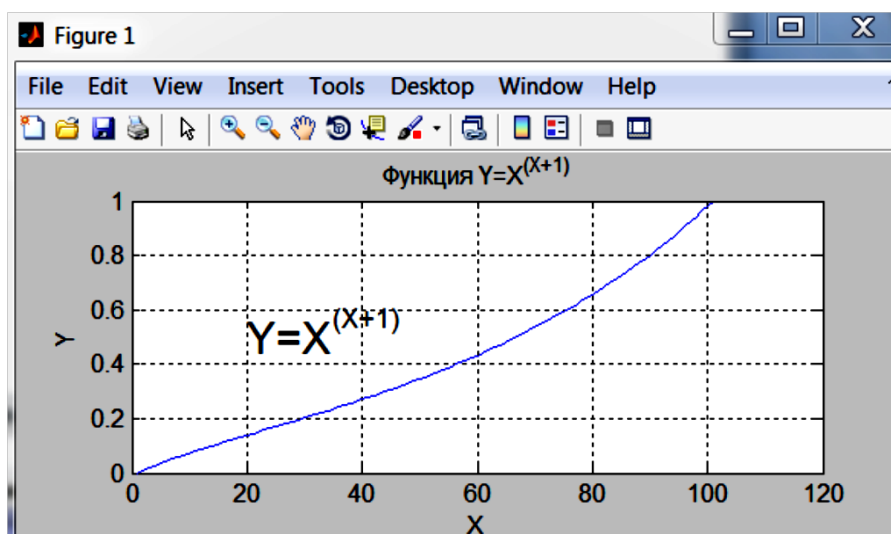


Рисунок 1.3.6

Координаты точки ввода текста в график удобно задавать с помощью мыши, используя команду **gtext('string')**. При исполнении этой команды на графике появляется большое перекрестие, установив которое в нужное место графика, достаточно нажать любую клавишу или любую кнопку мыши, и на этом месте появится надпись.

Если при неявном указании аргументов команды **plot** вектор функции **Y** содержит комплексные элементы (рис. 1.3.7), то при выполнении команды **plot(Y)** в качестве аргументов функции **Y** берется вещественная часть ее элементов, а в качестве значений функции мнимая часть ее элементов. Во всех других случаях мнимая часть данных игнорируется.

Результаты выполнения данного программного кода (см. рис. 1.3.7) приведены на рисунке 1.3.8.

```
Command Window
>> Y=[1+i*1 3+i*4];
>> plot(Y);grid
fx >>
```

Рисунок 1.3.7

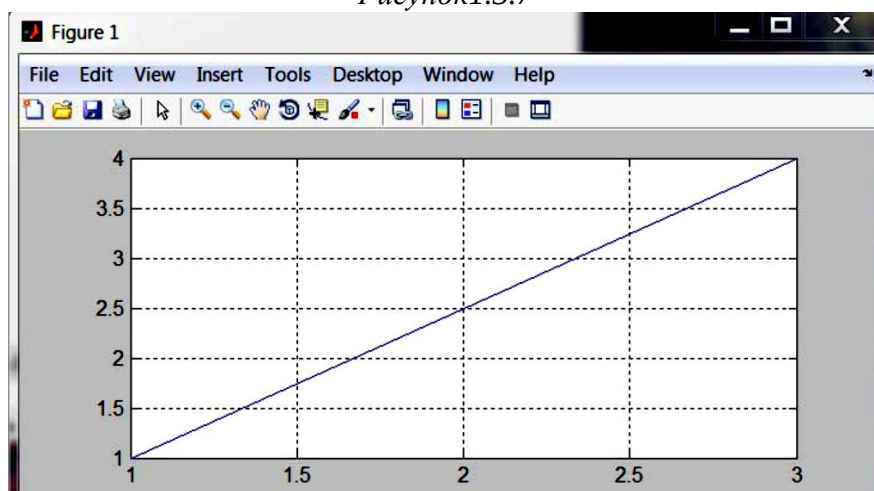


Рисунок 1.3.8

Трехмерная графика

Система MATLAB имеет богатые возможности построения трехмерных графиков. Рассмотрим только несколько функций, позволяющих создавать трехмерные графики.

Для создания трехмерного графика $z = f(x, y)$ необходимо иметь матрицы значений переменных x, y .

Для этого предназначены следующие функции:

[X, Y] = meshgrid(x, y)

[X, Y] = meshgrid(x)

[X, Y, Z] = meshgrid(x, y, z)

Функция **[X, Y] = meshgrid(x, y)** — преобразует область векторов x, y в массивы X, Y , которые используются для вычисления функции $z = f(x, y)$ и построения графиков.

Строки массива X являются копиями вектора x , а столбцы массива Y — копиями вектора y .

Функция **[X, Y, Z] = meshgrid (x, y, z)** возвращает трехмерный массив для построения трехмерного графика.

Графики трехмерных поверхностей строятся с помощью следующих функций:

plot3 (x, y, z)

plot3 (X, Y, Z)

plot3 (X, Y, Z, s)

plot3 (x1, y1, z1, s1, x2, y2, z2, s2, ..., xn, yn, zn, sn)

где:

- **x, y, z** — векторы аргументов функции;
- **X, Y, Z** — матрицы одинакового размера;
- **s** — стили линий и точек графика, аналогично функции **plot ()**.

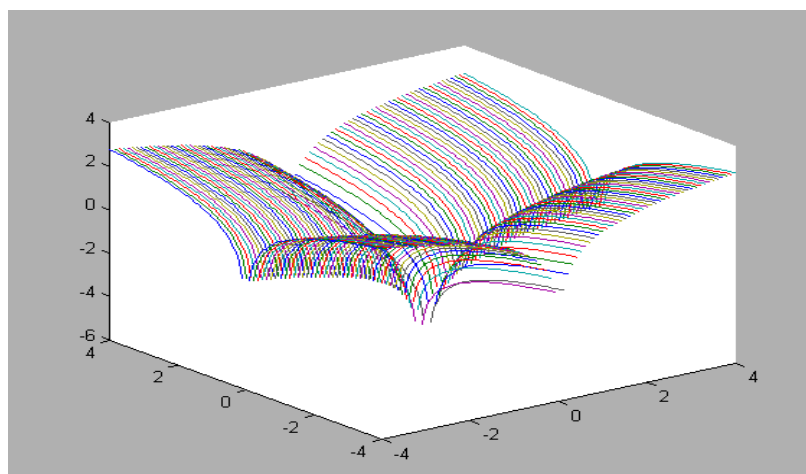
Приведенные функции строят точки трехмерного графика и соединяют их отрезками прямых в соответствии с заданным стилем.

Функция **plot3(x1, y1, z1, s1, x2, y2, z2, s2, ..., xn, yn, zn, sn)** строит на одном рисунке **n** функций.

Пример 1: построить график функции **$z = \ln x + \ln y$** в диапазоне аргументов **[-4; 4]** с шагом **h = 0.1**.

Command Window

```
>>[X,Y] = meshgrid([-4:0.1:4]);  
Z = log (X) + log (Y);  
plot3(X,Y,Z)
```

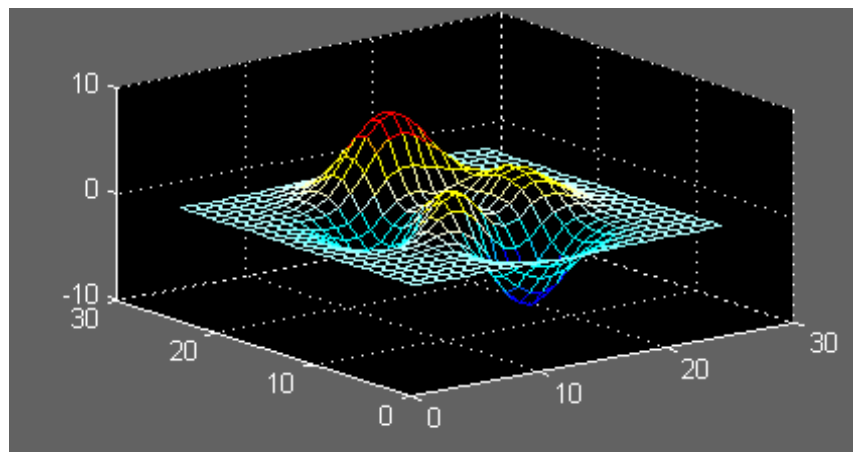


В Matlab существуют функции, специально предназначенные для формирования каркаса поверхности и самой поверхности по заданному массиву данных.

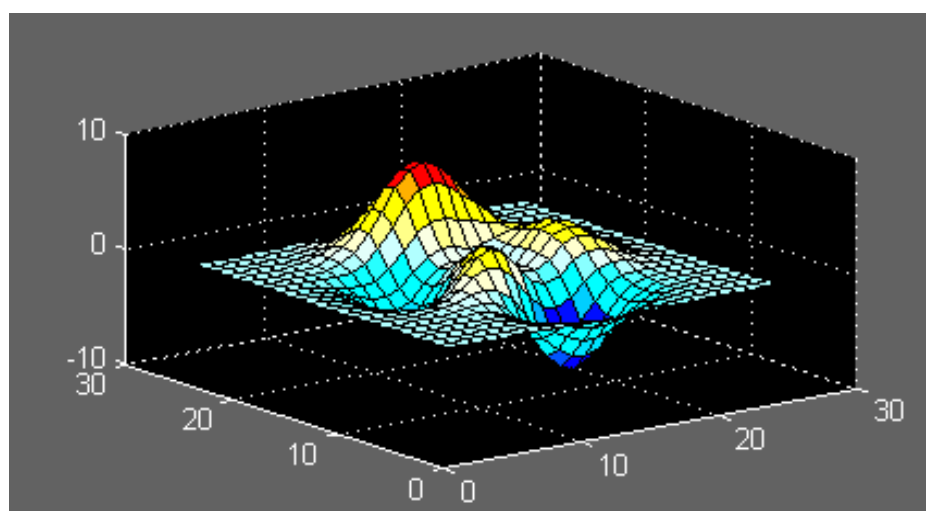
mesh (Z),

surf (Z), где **Z** — массив данных заданный матрицей.

Пример 2:
z=peaks (25);
mesh(z);



Пример 3:
z=peaks (25);
surf(z);



ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ № 3

Тема: «Графические построения в среде MATLAB»

Задание 1. Построить график функции для заданного варианта задания (табл. 1.3.4).

Номер варианта - последняя цифра в зачётной книжке.

Добавить к графику сетку из координатных линий, названия осей и заголовков.

Таблица 1.3.4

Номер варианта	Функция	Диапазон изменения x	Шаг изменения x
1	$Y = -\sin(x/2)$	$0^\circ - 360^\circ$	10°
2	$Y = 13x^2$	0 - 24	8
3	$Y = \cos(4x - 60^\circ)$	$0^\circ - 180^\circ$	5°
4	$Y = (3x^2 + 43)$	0 - 40	4
5	$Y = \operatorname{tg}(x) - 1$	$0^\circ - 90^\circ$	2°
6	$Y = (x^3 - x)$	0 - 200	20
7	$Y = \sin(4x - 45^\circ)$	$0^\circ - 180^\circ$	2°
8	$Y = \sin(3x + 60^\circ)$	$0^\circ - 360^\circ$	4°
9	$Y = 1 - e^{5x}$	0 - 5	0.1
10	$Y = \sin(5x)$	$0^\circ - 360^\circ$	4°

Задание 2. Построить график функции $z = \sin x + 2\cos y$ в диапазоне аргументов $[-3; 3]$ с шагом $h = 0.05$.

Выполнить построение каркаса поверхности и самой поверхности. Исходными данными является матрица (5×5) из случайных чисел, равномерно распределенных в диапазоне от 0 до 1

Включить в отчет по лабораторной работе содержимое командного окна и полученный график функции.

Оформить отчёт к лабораторной работе в виде файла и документа.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Содержание каждого задания.
4. Текст реализации в MATLAB с комментариями (комментарии отделяются знаком %).
5. Выводы по работе.

1.4 Лабораторная работа № 4

Тема: «Режим программирования в Matlab»

Цель работы:

1. Ознакомление с системой научных и инженерных расчетов Matlab;
- 2 . Ознакомление с программированием в среде MATLAB и созданием М-файлов.

Постановка задачи: используя теоретические сведения, выполнить задания лабораторной работы и подготовить отчет.

Краткие теоретические сведения

Программирование в среде MATLAB. Создание М-файлов.

Работа в режиме калькулятора в среде MatLAB, несмотря на довольно значительные возможности, имеет существенные недостатки. Невозможно повторить все предыдущие вычисления и действия при новых значениях исходных данных без повторного набора всех предыдущих операторов. Нельзя вернуться назад и повторить некоторые действия или по некоторому условию перейти к выполнению другой последовательности операторов. И вообще, если число операторов велико, становится проблемой отладить правильную их работу из-за неизбежных ошибок при наборе команд.

Поэтому сложные, с прерываниями, запутанными переходами по определенным условиям, с часто повторяемыми однотипными действиями вычисления, которые, к тому же, необходимо проводить неоднократно при измененных входных данных, требуют их специального оформления в виде записанных на диске файлов, т.е. в виде программ.

Преимущество программ в том, что становится возможным неоднократное обращение к одним и тем же операторам и к программе в целом. Создание программ позволяет значительно упростить и сократить процесс подготовки повторяемых вычислений, сделать процесс вычислений более наглядным и прозрачным. Кроме того, в программах появляется возможность автоматизировать процесс изменения значений входных параметров в диалоговом режиме.

В языке MATLAB имеются программы 2 типов:

- Script-файлы (управляющие программы);
- файл - функции (процедуры).

При помощи **Script-файлов** оформляются основные программы, управляющие от начала до конца организацией вычислительного процесса.

Файл - функции оформляются в отдельные процедуры и функции (т.е. такие части программы, которые рассчитаны на неоднократное использование при изменяемых входных параметрах). Файл-функция (процедура) должна начинаться со строки заголовка: ***function [<ПКВ>]=<имя процедуры>(<ПВВ>)***, где ПКВ - перечень конечных величин,

ПВВ - перечень входных величин.

Данные программы называются **М-файлами**. Все М-файлы имеют расширение *.m. М-файл должен храниться в директории которая является текущей. Для создания нового М-файла необходимо вызвать

File->New->M-file.

Пример M-файла.

function $y = F1(x, a)$

оператор **switch** $y = x + a/x$

Затем необходимо сохранить данный файл с именем F1.m

Так, если ввести в MATLAB команду $y = F1(1,2)$, то $y = 3$. **Выход из функции** в вызывающую программу обеспечивается выполнением последнего ее оператора или командой **return**.

MATLAB имеет 5 видов структур управления обработки данных – операторов:

1. Оператор **if**
2. Оператор **switch**
3. Оператор **for**
4. Оператор **while**
5. Оператор **break**

Оператор **if**

Условный оператор выступает в одной из следующих форм:

if <условие>
<команды>
end

if <условие>
<команды>
else
<команды>
end

if <условие>
<команды>
elseif <условие>
<команды>
else
<команды>
end

В роли условия может использоваться любое логическое выражение, построенное на основе операций отношения и логических операций. Если значение этого выражения является массивом, то условие считается истинным, если все его элементы истинны (истина – 1, ложь – 0).

Пример 1.

```
if rem(n,2) == 0
    M = odd_magic(n)
else
    M = double_even_magic(n)
end
```

Пример 2:

```
if A > B
    'greater'
elseif A < B
    'less'
elseif A == B
    'equal'
else
    error('Непредвиденная ситуация')
end
```

Оператор *switch*

Оператор переключения обобщает условный оператор на случай более двух условий и имеет конструкцию:

```
switch <выражение>
  case <значение 1>
    <команды>
  case <значение 2>
    <команды>
  ...
  otherwise                                % может отсутствовать
    <команды>
end
```

Пример:

```
switch k
  case 0
    t=1
  case (1, 2,5)
    t=2
  otherwise
    t=0
end
```

Оператор *for*

Оператор цикла с заданным числом повторений, в основном используемый в форме:

for V=A:H:B	for V=A:B
<команды>	<команды>
end	end

где V – переменная/параметр цикла, A, B – начальное и конечные значения; H – приращение, по умолчанию 1.

Допускаются и вложенные циклы. В заголовке цикла можно использовать одномерный массив.

Пример 1:

```
for i=1:n
  for j=1:m
    a(i,j)=x(i)^j;
  end
end
```

Пример 2:

```
for i=1:n-1
  for k=i+1:n
    if a(i)<a(k)
      m=a(i)
      a(i)=a(k)
      a(k)=m
    end
  end
```

```
end
end
end
```

Пример 3:

```
k=1;
for i=[0 5 7]
    x(k)=2^i;
    k=k+1;
end;
```

Оператор *while*

Оператор цикла с предусловием имеет традиционную конструкцию:

```
while <условие>
    <команды>
end
```

и обеспечивает выполнение команд тела цикла, пока истинно проверяемое условие.

Пример:

```
while a<1
    n=n+1
end
```

Оператор *break*

Работа любого цикла может быть прервана (выход из внутреннего или внешнего цикла) оператором **break**:

Пример:

```
while a<1
    n=n+1
    if n>250
        break
    end
```

Кроме упомянутых основных операторов, традиционных для любой системы программирования, в **MATLAB** существуют операторы обеспечения пользовательского интерфейса. **Ввод с клавиатуры** реализуется командой вида **<переменная>= input ('подсказка')**

Пример:

```
x=input(' степень полинома ');
Вывод имеет вид          степень полинома 3
```

Приостановка выполнения программы может быть предусмотрена включением в текст команды: **pause** (приостановка до нажатия любой клавиши), **pause (n)** (приостановка на n сек), **keyboard** (приостановка с возможностью выполнять практически любые команды и последующим возвратом в программу командой **return**).

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ № 4

Тема: «Режим программирования в Matlab»

Задание 1. Имеются 3 переменные a, b и i.

b=15. Переменная a=i/2. Переменная i=1.

На каждом шаге i увеличивается на 1.

Определить число шагов, за которое, a достигнет большего, чем b значения.

Задание 2. Имеются 2 переменные n и m.

Переменная n может принимать одно из двух значений 0 (m=n) или 1 (m=n+n/2)

Используя оператор переключения для переменной n, определить значение переменной m в каждом из этих случаев.

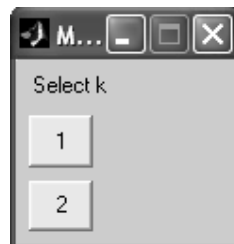
Задание 3. Дана матрица размером n×m. Произвести суммирование всех элементов матрицы кроме элементов последнего столбца и последней строки (используя вложенные циклы).

Задание 4. Создать функцию зависимости y от k и t.

Если k=1, то $y=(k*t)/2$

Если k=2, то y=0

Создать меню выбора значения переменной k (k может принимать значения 1 или 2). Выбирать одно из значений.



Вызвать М-файл, хранящий функцию. Значение параметра t=1.

Задание 5. Создать М-файл, реализующий вычисление следующей функции

$$y = d^3 * ctg(x) * \sqrt{\sin^4(x) - \cos^4(x)}$$

Оформить отчёт к лабораторной работе в виде файла и документа.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Содержание каждого задания.
4. Текст реализации в MATLAB с комментариями (комментарии отделяются знаком %).
5. Выводы по работе.

1.5 Лабораторная работа № 5

Тема: «Интегрирование функций
и решение дифференциальных уравнений»

Цель работы:

1. Ознакомление с системой научных и инженерных расчетов Matlab;
2. Научиться использовать встроенные функции для анализа функций в MATLAB.

Постановка задачи: используя теоретические сведения, выполнить задания лабораторной работы и подготовить отчет.

Краткие теоретические сведения

Интегрирование функций

В MATLAB имеется возможность вычисления определенных интегралов и двойных определенных интегралов с заданной точностью. При этом можно выбирать наиболее подходящий метод численного интегрирования.

Для вычисления определенного интеграла применяется функция **quad**. Ее первый аргумент – интегрируемая функция, второй и третий аргументы – нижний и верхний пределы интегрирования.

Пример 1.

$$\int_{-1}^1 e^{-x} \sin x dx$$

Текст программы:

```
function integral  
I=quad(@fint,-1,1)  
function f=fint(x)  
f=exp(-x).*sin(x)
```

Результат вычисления:

```
I =  
-0.663491467853105
```

По умолчанию задана точность 10-3. Если необходимо повысить точность, то нужно задать дополнительный четвертый аргумент функции **quad**.

```
I1=quad(@fint,-1,1, 1.0e-9)  
I1 =  
-0.663493666622457
```

Двойные интегралы вычисляются с помощью функции **dblquad**.

Пример 2.

$$\int_0^1 \int_{-\pi}^{\pi} (e^x \sin y + e^{-x} \cos y) dx dy$$

0 π

Текст программы:

```
function integral
I2=dblquad(@fint2,-pi, pi, 0,1)
function f=fint2(x,y)
f=exp(-x).*sin(y).^2+exp(-x).*cos(y).^2;
```

Результат вычислений:

I2 =
23.097478731539596

Решение дифференциальных уравнений

Для решения обыкновенных дифференциальных уравнений произвольного порядка и систем с начальными условиями (задач Коши) предназначены специальные встроенные функции-солверы. MATLAB имеет достаточно большой выбор солверов, реализующих различные методы. Пользователю необходимо выбрать подходящий солвер в зависимости от свойств решаемой задачи и произвести необходимые установки, обеспечивающие получение решения с заданной точностью.

Какие солверы используются, можно посмотреть в справочной системе MATLAB.

ODE solvers: ode45 ode23, ode113, ode15s, ode23s, ode23t, ode23tb

implicit ODEs: ode15i

options handling: odeset, odeget

output functions: odeplot, odephas2, odephas3, odeprint

evaluating solution: deval

ODE examples: rigidode, ballode, orbitode

function handles: function_handle

Схема решения состоит из следующих этапов:

1. Приведение дифференциального уравнения к системе дифференциальных уравнений первого порядка.
2. Написание файл-функции для системы уравнений.
3. Вызов подходящего солвера.
4. Визуализация результата.

Пример

$$y'' + 2y' + 10y = \sin(t),$$

начальные условия $y(0)=1, y'(0)=0$.

Вводим вспомогательные функции $y1$ и $y2$.

$$y1=y \quad y2=y'$$

$$\begin{cases} y1'=y2 \\ y2' = -2y2 - 10y1 + \sin t \end{cases} \quad \begin{bmatrix} y1(0) \\ y2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Текст программы:

```
function diffur
% начальные условия
Y0=[1;0];
% вызов солвера от созданной файл-функции,
% задание начального и конечного момента времени и % вектора
начальных условий
[T, Y]= ode45(@func,[0 15], Y0);
% вывод графика решения исходного дифференциального
уравнения plot(T, Y(:,1),'r')
hold on
% вывод на график производной
plot (T, Y(:,2),'k--')
grid on
function F=func(t,y)
F=[y(2); -2*y(2)-
10*y(1)+sin(t)];
```

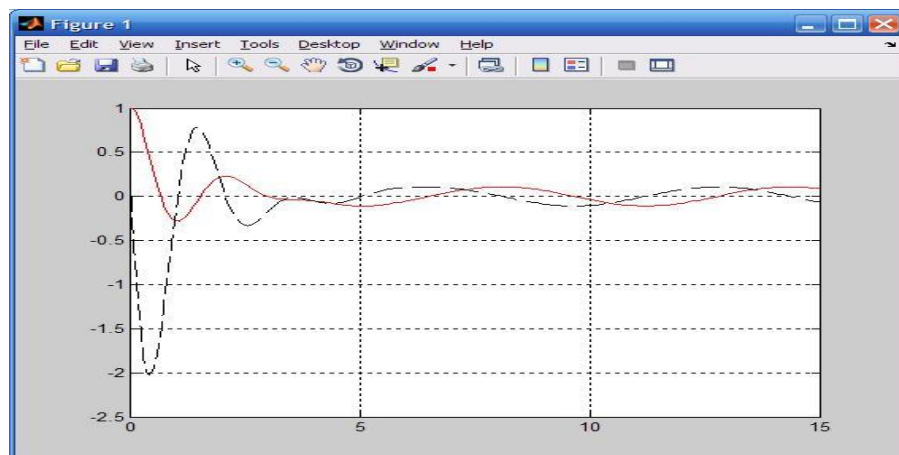


Рисунок 1.5.1

ЗАДАНИЕ К ЛАБОРАТОРНОЙ РАБОТЕ № 5

Тема: «Интегрирование функций и решение дифференциальных уравнений»

Задание 1. Интегрирование функций.

Вычислить определенный интеграл, предварительно задав необходимую функцию

$$\int_1^2 (x^2 - 2x + 3) dx$$

Задание 2. С применением интегрального исчисления вычислить и представить графически следующие задачи

Вычислить площадь, ограниченную параболой $y=4x-x^2$ и осью абсцисс

Задание 3. Решение задачи Коши.

1. Приведение дифференциального уравнения к системе дифференциальных уравнений первого порядка.

2. Написание специальной файл-функции для системы уравнений.

3. Вызов подходящего солвера.

4. Визуализация результата.

$$(1+x^2)y''-2xy'=0 \quad y=0 \quad y'=3 \quad \text{при } x=1$$

Оформить отчёт к лабораторной работе в виде файла и документа.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Содержание каждого задания.
4. Текст реализации в MATLAB с комментариями (комментарии отделяются знаком %).
5. Выводы по работе.

Глава 2. Практические занятия

2.1 Практическое занятие № 1

Тема: «Расчёт и моделирование электрической цепи переменного тока при последовательном соединении R, L, C»

Цель: Сформировать умения и навыки по моделированию электрической цепи переменного тока. Повторить алгоритм символического метода расчёта электрической цепи переменного тока, закрепить ранее полученные знания путём решения задачи по вариантам. Закрепить умение анализировать, полученные результаты расчёта.

Время: 2 часа.

Постановка задачи. Выполнить расчёт однофазной электрической цепи переменного тока, символическим методом, построить графики мгновенных значений тока и напряжений в среде Math Cad. Смоделировать расчётную электрическую цепь в среде MATLAB Simulink. Провести сравнительный анализ, полученных результатов. Сделать выводы.

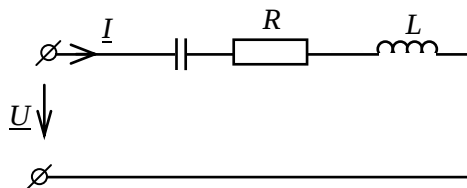


Рисунок 2.1.1

Задача. Индуктивная катушка с параметрами $R = 100$ Ом, $L = 5,05$ мГн и конденсатор $C = \dots$ мкФ соединены последовательно (рисунок 2.1.1). Действующее значение входного напряжения $U = \dots$ В.

Три последних цифры номера зачетной книжки – величина переменного напряжения, две последних – частота переменного напряжения. Число и месяц рождения – емкость фильтра в мкФ (пример 02.04.2000 – 204 мкФ).

Выполнить построение всех графиков в программной среде MATLAB SIMULINK.

Ссылка на видео для просмотра.

<https://www.youtube.com/watch?v=xPpLdIwJKmQ&t=538s>

Оформить отчёт в виде файла и документа.

2.2. Практическое занятие № 2

Тема: «Расчёт и моделирование электрической цепи переменного тока при параллельном соединении R , L , C »

Цель: Сформировать умения и навыки по моделированию электрической цепи переменного тока. Повторить алгоритм символического метода расчёта электрической цепи переменного тока, закрепить ранее полученные знания путём решения задачи по вариантам. Закрепить умение анализировать, полученные результаты расчёта.

Время: 2 часа.

Постановка задачи. Выполнить расчёт однофазной электрической цепи переменного тока, символическим методом, построить графики мгновенных значений токов и напряжения в среде Math Cad. Смоделировать расчётную электрическую цепь в среде MATLAB Simulink. Провести сравнительный анализ, полученных результатов. Сделать выводы.

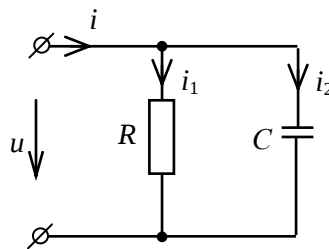


Рисунок. 2.2.1

Задание. В схеме рисунка 2.1.1. $R = 50 \text{ Ом}$, $C = \dots \text{ мкФ}$, $U = \dots \text{ В}$, $f = \dots \text{ Гц}$.

Три последних цифры номера зачетной книжки – величина переменного напряжения, две последних – частота переменного напряжения. Число и месяц рождения – емкость фильтра в мкФ (пример 02.04.2000 – 204 мкФ).

Выполнить построение всех графиков в программной среде MATLAB SIMULINK.

Ссылка на видео для просмотра

<https://www.youtube.com/watch?v=xPpLdIwJKmQ&t=594s>

Оформить отчёт в виде файла и документа.

2.3 Практическое занятие № 3

Тема: «Расчёт трёхфазных электрических цепей с применением среды MATLAB SIMULINK»

Цель: Ознакомление с системой научных и инженерных расчетов Matlab. Сформировать умения и навыки по моделированию электрической цепи переменного тока. Повторить алгоритм расчёта трёхфазной электрической цепи, закрепить раннее полученные знания путём решения задачи по вариантам. Закрепить умение анализировать, полученные результаты расчёта.

Время: 2 часа.

Постановка задачи: используя теоретические сведения, выполнить расчёт и моделирование трёхфазной электрической цепи и подготовить отчет в виде файла и документа.

Схема электрической цепи:

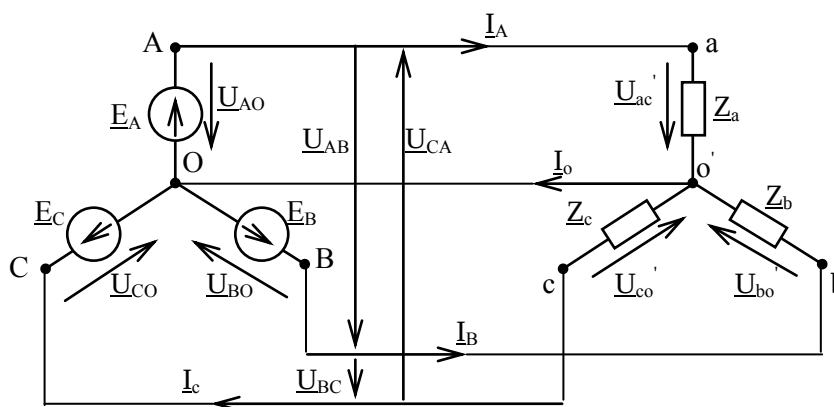


Рисунок. 2.3.1

Варианты задания:

1. К симметричному трехфазному генератору с линейным напряжением $U_\Delta = 380 \text{ В}$ присоединена нагрузка, состоящая из трех активных сопротивлений, соединенных звездой, $R = \dots$ (две последние цифры № зачетной книжки) Ом. Построить зависимости мгновенных значений от времени:

1. Линейных токов.
2. Линейных напряжений.
3. Фазных напряжений.

Построить векторную диаграмму. Выполнить построение все графиков в программной среде MATLAB SIMULINK.

2. К симметричному трехфазному генератору с линейным напряжением $U_\Delta = 380 \text{ В}$ присоединена нагрузка, состоящая из трех одинаковых катушек, соединенных звездой, $X_L = \dots$ (две последние цифры № зачетной книжки) Ом. Построить зависимости мгновенных значений от времени:

1. Линейных токов.
2. Линейных напряжений.

3. Фазных напряжений.

Построить векторную диаграмму. Выполнить построение все графиков в программной среде MATLAB SIMULINK.

3. К симметричному трехфазному генератору с линейным напряжением $U_{\Delta} = 380 \text{ В}$ присоединена нагрузка, состоящая из трех одинаковых емкостей, соединенных звездой, $X_c = \dots$ (две последние цифры № зачетной книжки) Ом. Построить зависимости мгновенных значений от времени:

1. Линейных токов.

2. Линейных напряжений.

3. Фазных напряжений.

Построить векторную диаграмму. Выполнить построение всех графиков в программной среде MATLAB SIMULINK.

Ссылка на видео для просмотра

<https://youtu.be/CDFtdhD80WM>

Оформить отчёт в виде файла и документа.

2.4 Практическое занятие № 4

Тема: «Расчёт и моделирование инвертора»

Цель: Сформировать умения и навыки по моделированию устройства преобразования переменного тока в постоянный - инвертора.

Время: 2 часа.

Постановка задачи: используя теоретические сведения, выполнить моделирование инвертора и подготовить отчет в виде файла и документа.

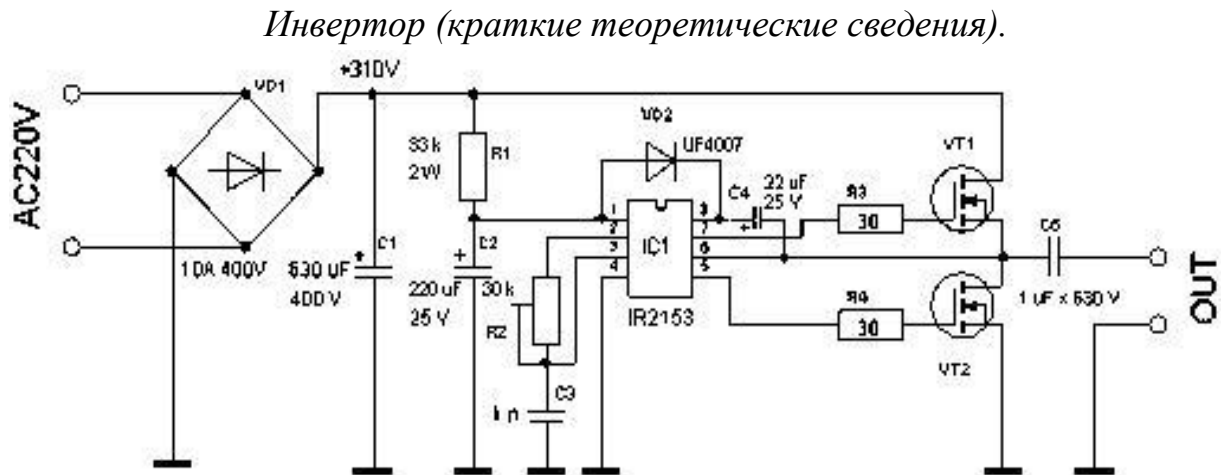


Рисунок 2.4.1. Электрическая схема инвертора

Устройства, предназначенные для преобразования постоянного тока в переменный называются **инверторами**.

Устройства преобразования переменного тока в постоянный характеризуются передачей активной мощности из питающей сети переменного тока в цепь постоянного тока – цепь нагрузки. В таких устройствах нередко требуется рекуперация энергии из цепи постоянного тока

в цепь переменного тока. В электроэнергетике это имеет место в передачах электроэнергии постоянным током. Подобная ситуация возникает и в тех случаях, когда выпрямительное устройство питает якорную цепь машины постоянного тока в системе электропривода какого-либо транспортного средства или грузоподъемного механизма. При движении транспорта под уклон или грузоподъемного механизма вниз (с грузом) машина постоянного тока переходит из двигательного режима работы в генераторный за счет механической энергии, подводимой к ней от исполнительного механизма. Эту энергию можно полезно использовать, преобразовав в электрическую и возвратив в цепь переменного тока.

Задание: Выполнить моделирование инвертора, по рассмотренному алгоритму в видео.

Три последних цифры номера зачетной книжки – величина постоянного напряжения, две последних – частота переменного напряжения.

Построить графики зависимости мгновенных значений от времени:

1. Входного напряжения.
2. Выходного напряжения.

Выполнить построение всех графиков в программной среде MATLAB SIMULINK.

Ссылка на видео для просмотра

<https://www.youtube.com/watch?v=FWQCgxLkBXA>

Оформить отчёт в виде файла и документа.

2.5 Практическое занятие № 5

Тема: «Расчёт и моделирование выпрямителя»

Цель: Сформировать умения и навыки по моделированию выпрямителя.

Время: 2 часа.

Постановка задачи: используя теоретические сведения, выполнить моделирование двухполупериодного (мостового) выпрямителя и подготовить отчет в виде файла и документа.

Выпрямитель (краткие теоретические сведения).

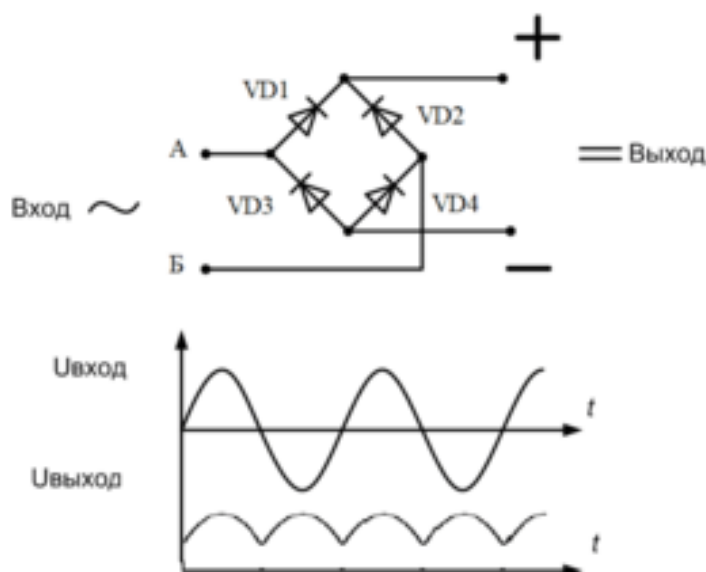


Рисунок.2.5.1. Выпрямитель

Для питания электронной аппаратуры, электродвигателей постоянного тока, электролизных и других установок возникает необходимость в выпрямлении переменного тока в постоянный.

Под **выпрямлением** понимается процесс преобразования переменного тока в постоянный с помощью устройств, обладающих односторонней проводимостью (электрических вентилей).

Выпрямительные устройства обычно состоят из трех основных элементов: трансформатора, электрического вентиля и сглаживающего фильтра. Трансформатор позволяет изменять значение переменного напряжения, получаемого от источника питания до значения требуемого выпрямленного напряжения.

Задание: Выполнить моделирование выпрямителя, преобразующего переменный ток в постоянный. Алгоритм выполнения просмотреть на видео.

Три последних цифры номера зачетной книжки – величина переменного напряжения, две последних – частота переменного

напряжения. Число и месяц рождения – емкость фильтра в мкФ (пример 02.04.2000 – 204 мкФ).

Построить зависимости мгновенных значений от времени:

1. Входного напряжения.
2. Выходного напряжения.

Выполнить построение всех графиков в программной среде MATLAB SIMULINK.

Ссылка на видео для просмотра

<https://www.youtube.com/watch?v=yJcKksL98Q>

Оформить отчёт в виде файла и документа.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Бессонов Л.А. Теоретические основы электротехники / Л.А. Бессонов. – М.: Высшая школа, 1973. – 724 с.
2. Зевеке Г.В. Основы теории цепей / Г.В.Зевеке, П.А.Ионкин, А.В. Нетушил, С.В. Страхов – М.: Энергоатомиздат, 1989. – 528 с.
3. Теоретические основы электротехники, Т1, 2. Учебник для вузов / К.С. Демирчан, Л.Р. Нейман, Н.В. Коровин, В.Л. Чечурин. – СПб: Питер, 2004.
4. Дьяконов В.П. MATLAB 6.5 SP1/7.0 + Simulink 5/6. Основы применения / В.П. Дьяконов. – Библиотека профессионала (Солон-пресс), книга #1,2,3, 2009. – 724 с.
5. Дьяконов В.П. MATLAB и Simulink в электроэнергетике: Справочник / В.П. Дьяконов. – Горячая линия – Телеком, 2009. – 816 с.

Учебное издание

**Абейдулин Сергей Адельзанович
Гайдук Светлана Владимировна
Скрябина Елена Викторовна**

**ПРИМЕНЕНИЕ ЭВМ
В ИНЖЕНЕРНЫХ РАСЧЕТАХ
И МОДЕЛИРОВАНИИ**

*Учебно-методическое пособие
для выполнения практических
и лабораторных работ по дисциплине*

В авторской редакции

Изд. № 91/2021. Объем 3 п.л.
РИИЦМ ФГАОУ ВО «Севастопольский государственный университет»