Министерство образования и науки РФ

Государственное образовательное учреждение

высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра вычислительной техники и систем управления

Методические указания к лабораторным работам по дисциплине

«Введение в MATLAB»

Владимир 2018

Министерство образования и науки РФ

Государственное образовательное учреждение

высшего профессионального образования

«Владимирский государственный университет имени

Александра Григорьевича и Николая Григорьевича Столетовых»

(ВлГУ)

Кафедра вычислительной техники и систем управления

Составители:

С.И. Лиходеев

Владимир 2018

УДК 621.396.218

ББК 32.884.1

М74

Рецензент

Кандидат педагогических наук, доцент кафедры

«Информатика и защита информации»

Владимирского государственного университета

Л. А.Артюшина

Представлено 8 лабораторных работ, которые выполняются в среде MATLAB. Предназначены для использования в лабораторных работах и практических занятиях студентами по дисциплине «Введение в MATLAB», направление подготовки 27.03.04 «Управление в технических системах» (бакалавриат).

# Введение

Система MATLAB предлагается разработчиками (корпорация MathWorks, Inc.) как лидирующий на рынке, в первую очередь на предприятиях военно-промышленного комплекса, в энергетике, в аэрокосмической отрасли и в автомобилестроении, язык программирования высокого уровня для технических вычислений, расширяемый большим числом пакетов прикладных программ — расширении. Она вобрала в себя не только передовой опыт развития и компьютерной реализации численных методов, накопленный за последние три десятилетия, но и весь опыт становления математики за всю историю человечества. [1]

Популярности системы MATLAB способствует ее матричная ориентация и мощное главное расширение Simulink. MATLAB и Simulink предоставляют пользователю удобные и простые средства, в том числе визуального объектно-ориентированного программирования, для моделирования линейных и нелинейных динамических систем, а также множество других пакетов расширения системы.

Начинающий пользователь MATLAB может в процессе работы совершенствовать свои знания как в области моделирования и численных методов, так и программирования и визуализации данных.

В данных методических указаниях рассмотрены 8 практических работ, для использования в лабораторных работах студентами по дисциплине «Введение в MATLAB», направление подготовки 27.03.04 «Управление в технических системах» (бакалавриат).

# Лабораторная работа № 1. Основы работы с MATLAB

**Цель занятия***:* изучение интерфейса пользователя системы MATLAB и основ работы с системой в режиме прямых вычислений.

* 1. **Основные теоретические сведения**

Исторически MATLAB разрабатывался как диалоговая среда для матричных вычислений (MATrix LABoratory). Со временем пакет был оснащен хорошей графической системой, дополнен средствами компьютерной алгебры от Maple и усилен библиотеками команд (или Toolboxes), предназначенными для эффективной работы со специальными классами задач.

В состав MATLAB входят интерпретатор команд, графическая оболочка, редактор-отладчик, библиотеки команд, компилятор, символьное ядро пакета Maple для проведения аналитических вычислений, математические библиотеки MATLAB на C/C++, генератор отчетов и богатый инструментарий (Toolboxes).

Интерфейс MATLAB вполне отвечает современным канонам (см. рисунок 1.1). Он многооконный и имеет ряд средств прямого доступа к различным компонентам системы. Следует обратить внимание на следующие кнопки панели инструментов:

**New M-file** — выводит пустое окно редактора m-файлов;

**Open file** — открывает окно для загрузки файлов Matlab;

**Simulink** — открывает окно браузера библиотек Simulink;

**Help** — открывает окно справки.

Эти функции дублируются в очень простом меню системы MATLAB.

В левой части окна системы появились окна со вкладками **Launch Pad/Workspace** доступа к компонентам системы и вкладками текущей директории **Current Directory** и истории сессии **History**. Они обеспечивают оперативный контроль за состоянием системы. Выводимые на экран окна интерфейса MATLAB могут быть включены или отключены из пункта меню View.

Вся работа организуется через командное окно (**Command Window**), которое появляется при запуске программы. В процессе работы данные располагаются в памяти (**Workspace**) в виде матриц.

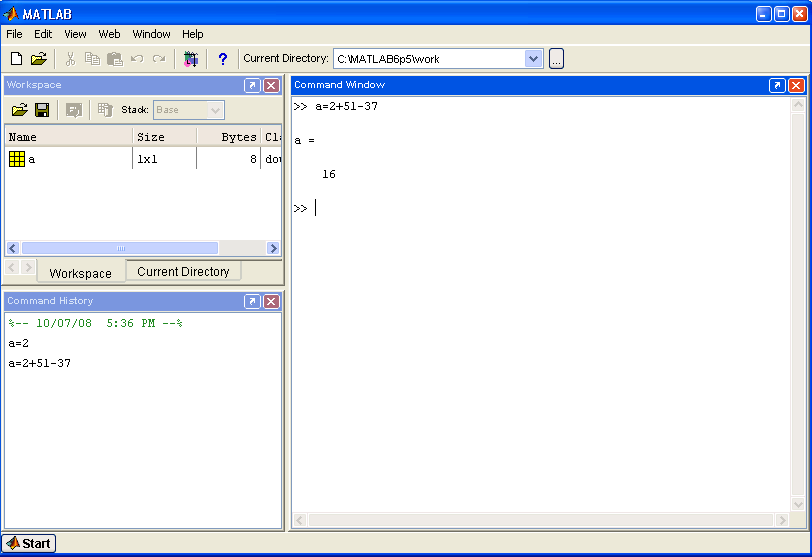


Рисунок 1.1 – Интерфейс программы Matlab

Все расчеты в MATLAB выполняются с двойной точностью, а для представления чисел на экране имеются разные форматы. Нужный формат может быть определен в меню (**File/Preferences**) либо при помощи команды **format**. Существуют следующие способы представления чисел (табл.1.1).

Таблица 1.1 Форматы вывода на экран

|  |  |
| --- | --- |
| **Формат** | **Представление** |
| short | Число отображается с 4 цифрами после десятичной точки или в формате short e |
| short e | Число в экспоненциальной форме с мантиссой из 5 цифр и показателем из 3 цифр |
| rat | Представление в виде рационального дробного числа |
| long | Число с 16 десятичными цифрами |
| long e | Число в экспоненциальной форме с мантиссой из 16 цифр и показателем из 3 цифр |
| hex | Число в шестнадцатеричной форме |

Переменные в MATLAB не нужно предварительно описывать, указывая их тип. Все данные хранятся в виде массивов: числовые переменные (внутренний тип numeriс), текстовые строки (char), ячейки (сеll) и структуры (struct). Двумерный массив – это матрица, одномерный – вектор, а скаляр – матрица размера 1x1. Имя переменной должно начинаться с буквы, за ней могут идти буквы, цифры и символ подчеркивания. Допустимы имена любой длины, но MATLAB идентифицирует их по первым 31 символам и различает большие и малые буквы. В MATLAB имеется ряд констант (табл.1.2).

Таблица 1.2 Зарезервированные имена констант

|  |  |
| --- | --- |
| **Имя** | **Описание** |
| ans | Результат последней операции |
| i, j | Мнимая единица |
| pi | Число π |
| eps | Машинная точность |
| realmax | Максимальное вещественное число |
| realmin | Минимальное вещественное число |
| inf | Бесконечность |
| NaN | Нечисловая переменная |
| end | Наибольшее значение индекса размерности массива |

Отметим, что имя NaN (Not-a-Number) зарезервировано для результата операций 0/0, 0\*inf, inf-inf и т.п.

Таблица 1.3 Специальные символы

|  |  |
| --- | --- |
| **Символ** | **Назначение** |
| [] | Квадратные скобки используются при задании матриц и векторов |
|  | Пробел служит для разделения элементов матриц |
| , | Запятая применяется для разделения элементов матриц и оператора в строке ввода |
| ; | Точка с запятой отделяет строки матриц, а точка с запятой в конце оператора (команды) отменяет вывод результата на экран |
| : | Двоеточие используется для указания диапазона (интервала изменения величины) и в качестве знака групповой операции над элементами матриц |
| () | Круглые скобки применяются для задания порядка выполнения математических операций, а также для указания аргументов функций и индексов матриц |
| . | Точка отделяет дробную часть числа от целой его части, а также применяется в составе комбинированных знаков (.\*, .^, ./, .\) |
| … | Три точки и более в конце строки отмечают продолжение выражения на следующей строчке |
| % | Знак процента означает начало комментария |
| ’ | Апостроф указывает на символьные строки, а для включения самого апострофа в символьную строку нужно поставить два апострофа подряд |

В командном окне в режиме диалога проводятся вычисления. Пользователь вводит команды или запускает на выполнение файлы с текстами на языке

MATLAB. Интерпретатор обрабатывает введенное значение и выдает результаты: числовые и строковые данные, предупреждения и сообщения об ошибках. Строка ввода помечена знаком **>>**.

При работе с MATLAB в командном режиме действует простейший строчный редактор. Обратите особое внимание на применение клавиш **Up** и **Down** (стрелки курсора "Вверх" и "Вниз"). Они используются для подстановки после маркера строки ввода >> ранее введенных строк из специального стека, например, для их исправления, дублирования или дополнения. При этом указанные клавиши обеспечивают перелистывание ранее введенных строк снизу вверх или сверху вниз.

Имена переменных должны начинаться с буквы. Знак = соответствует операции присваивания. Нажатие клавиши *Enter* заставляет систему вычислить выражение и показать результат. Если запись оператора не заканчивается символом «;», то результат выводится в командное окно, в противном случае – не выводится. Если оператор не содержит знака присваивания «=», то значение результата присваивается системной переменной *ans* (см. рисунок 1.2).

Все значения переменных, вычисленные в течение текущего сеанса работы, сохраняются в специально зарезервированной области памяти компьютера, называемой рабочим пространством системы MATLAB (**Workspace**).

Для просмотра значения любой переменной из текущего рабочего пространства системы достаточно набрать ее имя и нажать клавишу Enter.

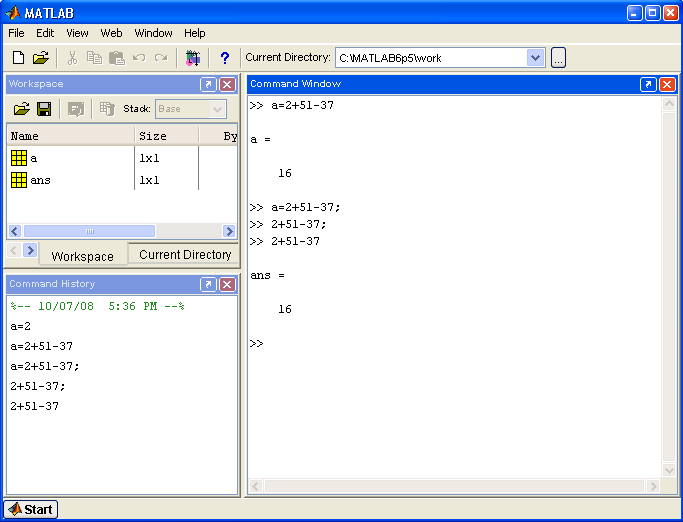


Рисунок 1.2 – Демонстрация выполнения команды присваивания

После окончания сеанса работы с системой MATLAB все ранее вычисленные переменные теряются. Чтобы сохранить в файле на диске компьютера содержимое рабочего пространства системы MATLAB, нужно выполнить команду меню ***File* \*Save Workspace As*** … .По умолчанию расширение имени файла mat, поэтому такие файлы принято называть МАТ-файлами.

Система MATLAB работает как с действительными, так и с комплексными числами. Перед использованием операций с комплексными числами необходимо определить переменную *i* = sqrt(–1) или *j* = sqrt(–1). В арифметических выражениях применяются следующие знаки операций:

+, - – сложение, вычитание,

\*– умножение,

/ – деление слева направо;

\ – деление справа налево;

^ – возведение в степень.

Система MATLAB позволяет вычислять различные математические функции. Следующие элементарные алгебраические функции имеют в качестве аргумента одно или два действительных (*x*, *y*) или одно комплексное (*z*) число (табл. 1.4).

Таблица 1.4 Элементарные алгебраические функции

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **abs**(*z*), **abs**(*x*), | Вычисление модуля комплексного числа *z* или абсолютного значения действительного числа *x.* |
| **angle**(*z*) | Вычисление аргумента *z.* |
| **sqrt**(*z*), **sqrt**(*x*) | Вычисление квадратного корня чисел *z* и *x* |
| **real**(*z*) | Вычисление действительной части комплексного числа *z.* |
| **imag**(*z*) | Вычисление мнимой части комплексного числа *z.* |
| **round**(*x*) | Округление до целого. |
| **fix**(*x*) | Округление до ближайшего целого в сторону нуля. |
| **rem**(*x*, *y*) | Вычисление остатка от деления *x* на *y*. |
| **exp(***z*) | Вычисление *е* в степени *x.* |
| **log**(*z*) | Вычисление натурального логарифма числа *x.* |
| **log10**(*z*) | Вычисление десятичного логарифма числа *x*. |

Система MATLAB предоставляет возможности для вычисления следующих тригонометрических и обратных тригонометрических функций переменной *x* (табл.1.5).

Таблица 1.5 Тригонометрических функций

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

**1.2 Порядок выполнения**

1. В командном окне задать значения переменных, согласно варианту задания, представленному в таблице 1.6.
2. Записать выражение на языке MATLAB.

Таблица 1.6 Варианты заданий

|  |  |  |
| --- | --- | --- |
| **№** | **Выражение** | **Переменные** |
| **1** | **2** | **3** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
| 9 |  |  |
| 10 |  |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 |  |  |
| 15 |  |  |

**1.3 Содержание отчета**

1. Цель работы.

2. Пример расчета и вывода данных.

**1.4 Контрольные вопросы**

1. Для чего служит команда HELP?

2. Перечислите основные команды MATLAB для работы в режиме прямых вычислений.

3. С помощью какой команды, устанавливается формат чисел?

4. Перечислите основные системные переменные MATLAB.

# Лабораторная работа № 2. Операции с векторами и матрицами в системе MATLAB

**Цель занятия***:* изучение реализации средствами системы MATLAB основных операций с векторами и матрицами.

**2.1 Основные теоретические сведения**

По умолчанию все числовые переменные в MATLAB считаются матрицами, так что скалярная величина есть матрица первого порядка, а векторы являются матрицами, состоящими из одного столбца или одной строки. Матрицу можно ввести, задав ее элементы или считав данные из файла, а также в результате обращения к стандартной или написанной пользователем функции.

Матричные данные размещаются в памяти последовательно по столбцам. Элементы матрицы в пределах строки отделяются пробелами или запятыми. Непосредственное задание матрицы можно осуществить несколькими способами. Например, вектор-столбец, то есть матрица, вторая размерность которой равна единице, может быть присвоена переменной *А* вводом одной строки:

>> A=[7+4i; 4; 3.2] % Ввод вектора-столбца

A =

7.0000 + 4.0000i

4.0000

3.2000

или вводом нескольких строк

>> A = [ % ввод вектора по строкам

7+4i

4

3.2];

Векторы могут быть сформированы как диапазоны – при помощи двоеточий, разделяющих стартовое значение, шаг и предельное значение. Если величина шага отсутствует, то по умолчанию его значение равно единице.

В результате *n*:*m*:*k* будет сформирован вектор, последний элемент которого не больше *k* для положительного шага *m*, и не меньше – для отрицательного: [*n*, *n*+*m*,*n*+*m*+*m*,…]

Например:

>> a=1:2:5

a =

1 3 5

Задание диапазона используется также при организации цикла.

В таблице 2.1 представлен некоторый набор функций для создания матриц специального вида.

Таблица 2.1. Функции описания матриц

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **eye**(m,n) | Единичная матрица размерности *m*×*n* |
| **zeros**(m,n) | Нулевая матрица размерности *m*×*n* |
| **ones**(m,n) | Матрица, состоящая из одних единиц размерности *m*×*n* |
| **rand**(m,n) | Возвращает матрицу случайных чисел равномерно распределенных в диапазоне от 0 до 1, размерность *m*×*n* |
| **randn**(m, n) | Возвращает матрицу размерности *m*×*n*, состоящих из случайных чисел, имеющих гаусовское распределение |
| **tril**(A), **triu**(A) | Выделение нижней треугольной и верхней треугольной частей матрицы *A* |
| **inv**(A) | Нахождение обратной матрицы *A* |
| **det**(A) | Нахождение определителя (детерминанта) квадратной матрицы *A* |

Обращение к элементу матрицы производится по правилу, – в круглых скобках после имени матрицы даются индексы, которые должны быть положительными целыми числами, указывающими номер строки и через запятую, номер столбца. Например, А(2,1) означает элемент из второй строки первого столбца матрицы А.

Для дальнейших примеров введем матрицу 2x2:

>> A=[1 2+5\*i; 4.6 3]

A =

1.0000 2.0000 + 5.0000i

4.6000 3.0000

Чтобы изменить элемент матрицы, ему нужно присвоить новое значение:

>> A(2,2)=10 % Второй элемент второй строки

A =

1.0000 2.0000 + 5.0000i

4.6000 10.0000

Размер матрицы можно уточнить по команде **size**, а результат команды **size** можно использовать для организации новой матрицы.

Например, нулевая матрица того же порядка, что и матрица *А*, будет сформирована по команде

>> A2=zeros(size(A))

A2 =

0 0

0 0

С помощью двоеточия легко выделить часть матрицы. Например, вектор из первых двух элементов второго столбца матрицы *A* задаётся выражением:

>> A(1:2, 2)

ans =

2.0000 + 5.0000i

10.0000

Двоеточие само по себе означает строку или столбец целиком. Для удаления элемента вектора достаточно присвоить ему пустой массив – пару квадратных скобок []. Чтобы вычеркнуть одну или несколько строк (столбцов) матрицы нужно указать диапазон удаляемых строк (столбцов) для одной размерности и поставить двоеточие для другой размерности. Для нахождения длины вектора можно воспользоваться также командой **length**.

Набор арифметических операций в MATLAB для работы с матрицами состоит из стандартных операций сложения – вычитания, умножения – деления, операции возведения в степень и дополнены специальными матричными операциями (табл.2.2). Если операция применяется к матрицам, размеры которых не согласованы, то будет выведено сообщение об ошибке.

Для поэлементного выполнения операций умножения, деления и возведения в степень применяются комбинированные знаки (точка и знак операции). Например, если за матрицей стоит знак (^), то она возводится в степень, а комбинация (.^) означает возведение в степень каждого элемента матрицы. При умножении (сложении, вычитании, делении) матрицы на число соответствующая операция всегда производится поэлементно.

Таблица 2.2 Знаки операций

|  |  |
| --- | --- |
| **Символ** | **Назначение** |
| +,- | Символы плюс и минус обозначают знак числа или операцию сложения и вычитания матриц, причем матрицы должны быть одной размерности |
| \* | Знак умножения обозначает матричное умножение, для поэлементного умножения матрицы применяется комбинированный знак (.\*) |
| ' | Апостроф обозначает операцию транспонирования (вместе с комплексным сопряжением), транспонирование без вычисления сопряжения обозначается при помощи комбинированного знака (.') |
| / | Левое деление |
| \ | Правое деление |
| ^ | Оператор возведения в степень, для поэлементного возведения в степень применяется комбинированный знак (.^) |

Проиллюстрируем различие обычного и поэлементного умножений при помощи следующего примера.

Введём матрицу *H* размера 2х2 и матрицу *D* из единиц той же размерности:

>> H=[0 1; 2 3], D=ones(size(H))

H =

0 1

2 3

D =

1 1

1 1

Перемножим матрицы, используя обычное умножение:

>> H\*D

ans =

1 1

5 5

Теперь применим поэлементную операцию:

>> H.\*D

ans =

0 1

2 3

Система MATLAB имеет ряд функций, предназначенных для обработки данных, заданных в матричной или векторной форме (таблица 2.3).

Таблица 2.3 Функции для работы с матрицами

|  |  |
| --- | --- |
| **Функция** | **Описание** |
| **size**(A) | Возвращает массив, состоящий из числа строк и числа столбцов матрицы. |
| **sum**(A) | Возвращает сумму всех элементов по столбцу |
| **mean**(A) | Возвращает среднее значение столбца матрицы |
| **std**(A) | Возвращает среднеквадратическое отклонение столбца матрицы |
| **min**(A), **max**(A) | Возвращает минимум и максимум соответственно, по столбцу матрицы |
| **sort**(A) | Сортирует столбец матрицы по возрастанию |
| **prod**(A) | Вычисляет произведение всех элементов столбцов |

Символы и текстовые строки в MATLAB вводятся при помощи простых кавычек. Во внутреннем представлении символы даны целыми числами. Конвертировать массив символов в числовую матрицу позволяет команда **double**. Обратная операция совершается по команде **char**. Печатаемые символы из стандартного набора ASCII представлены числами от 32 до 255.

Приведем примеры для данных команд. Вначале введем строку:

**>>** s = 'Привет'

s =

Привет

Отметим, что для ввода русских букв следует выбрать в меню File/ Preferences/ Command Windows Font шрифт с русской кодировкой.

**>>** h = [v + ' от MATLAB']

v =

Привет от MATLAB

Тот же результат получится, если вместо переменной *v* использовать строковую переменную *s*.

Для перевода численных данных в строковые переменные имеется ряд команд преобразования. В таблице 2.4 приведены некоторые функции для этих и обратных операций, а полных список можно получить по команде **help** **strfun**.

Таблица 2.4 Функции работы со строковыми переменными

|  |  |
| --- | --- |
| Функция | Действие |
| **num2str** | Перевод числа в строку |
| **int2str** | Перевод целого числа в строку |
| **mat2str** | Преобразование матрицы в строку |
| **str2mat** | Объединение строк в матрицу |
| **str2num** | Преобразование строки в число |
| **strcat** | Объединение строк |

**2.2 Порядок выполнения**

1. Ввод с клавиатуры векторов и матриц.

Ввести:

– произвольную вектор-строку (*v*), размерность 2;

– произвольный вектор-столбец (*w*), размерность 2;

– произвольную матрицу (*m*), размерности 2×2.

2. Генерация матриц специального вида.

Создать:

– матрицу с нулевыми элементами (*m0*), размерности 2×2;

– матрицу с единичными элементами(*m1*), размерности 2×2;

– матрицу с элементами, имеющими случайные значения(*mr*), размерности 2×2;

– матрицу с единичными диагональными элементами(*me*), размерности 2×2.

3. Вычисление матрицы *M* по формуле, представленной в таблице 2.5.

4. Изучение функций обработки данных:

– определение числа строк и столбцов матрицы *M*;

– определение максимального элемента матрицы *M*;

– определение минимального элемента матрицы *M*;

– суммирование элементов матрицы *M*;

– перемножение элементов матрицы *M*.

Таблица 2.5 Варианты заданий

|  |  |  |  |
| --- | --- | --- | --- |
| **№ варианта** | **Задание** | **№ варианта** | **Задание** |
| 1 | *M=v\*w+m+mr\*me* | 11 | *M=m\*w+mr\*v’* |
| 2 | *M=m+mr\*me* | 12 | *M=m\*mr+w\*v* |
| 3 | *M=(v/m)\*(mr+me)* | 13 | *M=m+mr–100* |
| 4 | *M=w\*v+mr\*me* | 14 | *M=v’+w+mr\*w* |
| 5 | *M=m\*mr+me* | 15 | *M=m+m1’\*me’* |
| 6 | *M=m.\*mr+100* | 16 | *M=(v/m)\*(mr+me)* |
| 7 | *M=v\*w+mr–m* | 17 | *M=v\*mr+v\*m1* |
| 8 | *M=m+mr\*me–10* | 18 | *M=m’+mr/100* |
| 9 | *M=m\*w+mr\*v'* | 19 | *M=10\*v+w’\*mr\*m* |
| 10 | *M=m’+mr\*me* | 20 | *M=m’+mr\*me* |

**2.3 Содержание отчета**

1. Цель работы.

2. Описание ввода с клавиатуры векторов и матриц.

3. Описание команд генерации матриц специального вида.

4. Описание основных функций обработки данных.

**2.4 Контрольные вопросы**

1. Как осуществляется ввод вектора–строки?

2. Как осуществляется ввод вектора–столбца?

3. Как осуществляется ввод матрицы?

4. Для чего служит команды zeros, ones, rand, eye?

5. Как определяется число строк и столбцов матрицы?

6. Какие операции служат для определения минимального и максимального элемента матрицы?

# Лабораторная работа № 3. Решение систем уравнений в MATLAB

**Цель занятия***:* изучение возможностей решения системы уравнений графически.

3.1. Выбрать систему уравнений (Таб.3.1). Составить программу. Пример программы приведен ниже.

По полученным графикам определить корни системы уравнений и делать проверку

Таблица 3.1

|  |  |
| --- | --- |
| № варианта | Задание |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |

. > x1=-2:0.1:2;

y1=sin(x1-0.6)-1.6;

y2=-3:0.1:3;

x2=(0.9+cos(y2))/3;

plot(x1, y1, 'R', x2, y2)

grid on



**3.2 Содержание отчета**

1. Цель работы.

2. Программа построения графиков.

3. Значение найденных корней.

4. Выводы.

**3.3 Контрольные вопросы**

1. Как осуществляется построение графиков в Matlab?

2. Как отформатировать графики?

3. Как осуществляется проверка решения системы уравнений?

# Лабораторная работа № 4. Аппроксимация полиномами в среде MATLAB

**Цель занятия***:* Аппроксимация данных полиномами в MATLAB.

1. Аппроксимация полиномами в среде MATLAB осуществляется с помощью функции polyfit(), которая имеет вид:

polyfit(x, y, n)

где:

* *x* – вектор узлов интерполяции;
* *y* – вектор значений функции в узлах интерполяции;
* *n* – степень полинома.

Откликом при реализации функции polyfit() является вектор коэффициентов полинома. Рассмотрим пример использования функции polyfit.

**Пример**

Выполните следующие действия:

>> x=[0.43;0.48;0.55;0.62;0.70;0.75];

>> y=[1.6359;1.7323;1.8768;2.0304;2.2284;2.3597];

>> polyfit(x,y,5)

После нажатия клавиши <Enter> ответ получим в следующем виде:

ans =

-152.9063 444.9904 -511.6367 291.7494 -80.6863 10.0997

Тогда функцией интерполяции будет следующий полином пятой степени:



Пример программы и график функции приведен ниже.

Программа

x=(-3:1:3)' **% Задание х**

y=[-0.71 -0.01 0.51 0.82 0.88 0.51 0.49]'**% Задание y**

p=polyfit(x,y,6) **%вычислим коэффициенты аппроксимирующего полинома степени 6**

f = polyval(p,x); **%вычислим значения полинома в точках сетки**

table=[x y f y-f**]%формируем таблицу данных**

**%построим графики функции и аппроксимирующего полинома на отрезке [-3 3].**

x=(-3:1:3)';

f=polyval(p,x);

plot(x,y,'ob',x,f,'-g'),axis([-3 3 -1 3]),grid



Рис.4.1 График функции

Таблица с результатами расчетов.

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | f | у-f |
| -3.0000 | -0.7100 | --0.7100 | 0 |
| -2.0000 | -0.0100 | -0.0100 | -0.0000 |
| 1.0000 | -0.5100 | 0.5100 | 0.0000 |
| 0 | 0.8200 | 0.8200 | 0.0000 |
| 1.0000 | 0.8800 | 0.8800 | -0.0000 |
| 2.0000 | 0.5100 | 0.5100 | -0.0000 |
| 3.0000 | 0.4900 | 0.4900 | -0.0000 |

Варианты заданий приведены в таблице 4.1

Таблица 4.1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | | 2 | | 3 | |
| *x* | *y* | *x* | *y* | *x* | *y* |
| 1,415 | 0,8886 | 0,101 | 1,2618 | 0,15 | 0,8607 |
| 1,420 | 0,8900 | 0,106 | 1,2764 | 0,20 | 0,8187 |
| 1,425 | 0,8906 | 0,111 | 1,2912 | 0,25 | 0,7788 |
| 1,430 | 0,8917 | 0,116 | 1,3061 | 0,30 | 0,7408 |
| 1,435 | 0,8927 | 0,121 | 0,3213 | 0,35 | 0,7046 |
| 1,440 | 0,8940 | 0,126 | 1,3366 | 0,40 | 0,6703 |
| 1,445 | 0,8947 | 0,131 | 1,3521 | 0,45 | 0,6376 |
| 1,450 | 0,8957 | 0,136 | 1,3677 | 0,50 | 0,6065 |
| 1,455 | 0,8967 | 0,141 | 1,3836 | 0,55 | 0,5769 |
| 1,460 | 0,8977 | 0,146 | 1,3995 | 0,60 | 0,5488 |
| 1,465 | 0,8986 | 0,151 | 1,4157 | 0,65 | 0,5220 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | | 5 | | 6 | |
| *x* | *y* | *x* | *y* | *x* | *y* |
| 0,180 | 5,6154 | 3,50 | 33,115 | 0,115 | 8,6572 |
| 0,185 | 5,4669 | 3,55 | 34,813 | 0,120 | 8,2932 |
| 0,190 | 5,3263 | 3,60 | 36,598 | 0,125 | 7,9582 |
| 0,195 | 5,1930 | 3,65 | 38,474 | 0,130 | 7,6489 |
| 0,200 | 5,0664 | 3,70 | 40,447 | 0,135 | 7,3623 |
| 0,205 | 4,9461 | 3,75 | 42,521 | 0,140 | 7,0961 |
| 0,210 | 4,8317 | 3,80 | 44,701 | 0,145 | 6,8481 |
| 0,215 | 4,7226 | 3,85 | 46,993 | 0,150 | 6,6165 |
| 0,220 | 4,6185 | 3,90 | 49,402 | 0,155 | 6,3998 |
| 0,225 | 4,5191 | 3,95 | 51,935 | 0,160 | 6,1965 |
| 0,230 | 4,4242 | 4,00 | 54,598 | 0,165 | 6,0055 |
| 7 | | 8 | | 9 | |
| *x* | *y* | *x* | *y* | *x* | *y* |
| 1,340 | 4,2556 | 0,01 | 0,9918 | 0,15 | 4,481 |
| 1,345 | 4,3532 | 0,06 | 0,9519 | 0,16 | 4,953 |
| 1,350 | 4,4552 | 0,11 | 0,9136 | 0,17 | 5,473 |
| 1,355 | 4,5618 | 0,16 | 0,8769 | 0,18 | 6,049 |
| 1,360 | 4,6734 | 0,21 | 0,8416 | 0,19 | 6,685 |
| 1,365 | 4,7903 | 0,26 | 0,8077 | 0,20 | 7,389 |
| 1,370 | 4,9130 | 0,31 | 0,7753 | 0,21 | 8,166 |
| 1,375 | 5,0419 | 0,36 | 0,7441 | 0,22 | 9,025 |
| 1,380 | 5,1774 | 0,41 | 0,7141 | 0,23 | 9,974 |
| 1,385 | 5,3201 | 0,46 | 0,6854 | 0,24 | 11,023 |
| 1,390 | 5,4706 | 0,51 | 0,6579 | 0,25 | 12,182 |
| 10 | | 11 | | 12 | |
| *x* | *y* | *x* | *y* | *x* | *y* |
| 0,45 | 20,194 | 1,345 | 4,3532 | 0,130 | 7,6489 |
| 0,46 | 19,613 | 1,350 | 4,4552 | 0,135 | 7,3623 |
| 0,47 | 18,942 | 1,355 | 4,5618 | 0,140 | 7,0961 |
| 0,48 | 18,174 | 1,360 | 4,6734 | 0,145 | 6,8481 |
| 0,49 | 17,301 | 1,365 | 4,7903 | 0,150 | 6,6165 |
| 0,50 | 16,312 | 1,370 | 4,9130 | 0,155 | 6,3998 |
| 0,51 | 15,198 | 1,375 | 5,0419 | 0,160 | 6,1965 |
| 0,52 | 13,948 | 1,380 | 5,1774 | 0,165 | 6,0055 |
| 0,53 | 12,550 | 1,385 | 5,3201 | 0,170 | 5,8255 |
| 0,54 | 10,993 | 1,390 | 5,4706 | 0,175 | 5,6558 |
| 0,55 | 9,264 | 1,395 | 5,6296 | 0,180 | 5,4954 |

**4.2 Содержание отчета**

1. Цель работы.

2. Программа.

3. Результаты расчета.

4. Выводы.

**4.3 Контрольные вопросы**

1. Для чего служит функция polyfit(),?

2. Для чего служит функция polyval(p,x)?

3. Как напечатать в Matlab таблицу данных?

4. Какой функцией реализуется сплайн-интерполяция в MATLAB?

# Лабораторная работа № 5 Расчет цепи постоянного тока

*Цель работы: Создание модели и расчет токов и напряжения в цепи постоянного тока*

В данной работе необходимо построить модель электрической схемы представленной на рис 5.1 в пакете Simulink.

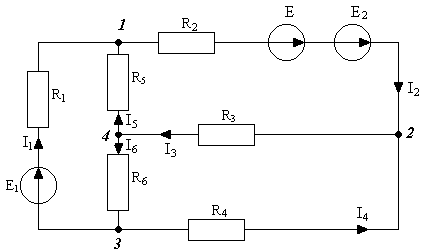


Рис 5.1

Пример модели представлен на рис 5.2. Варианты для расчета представлены в таблице 5

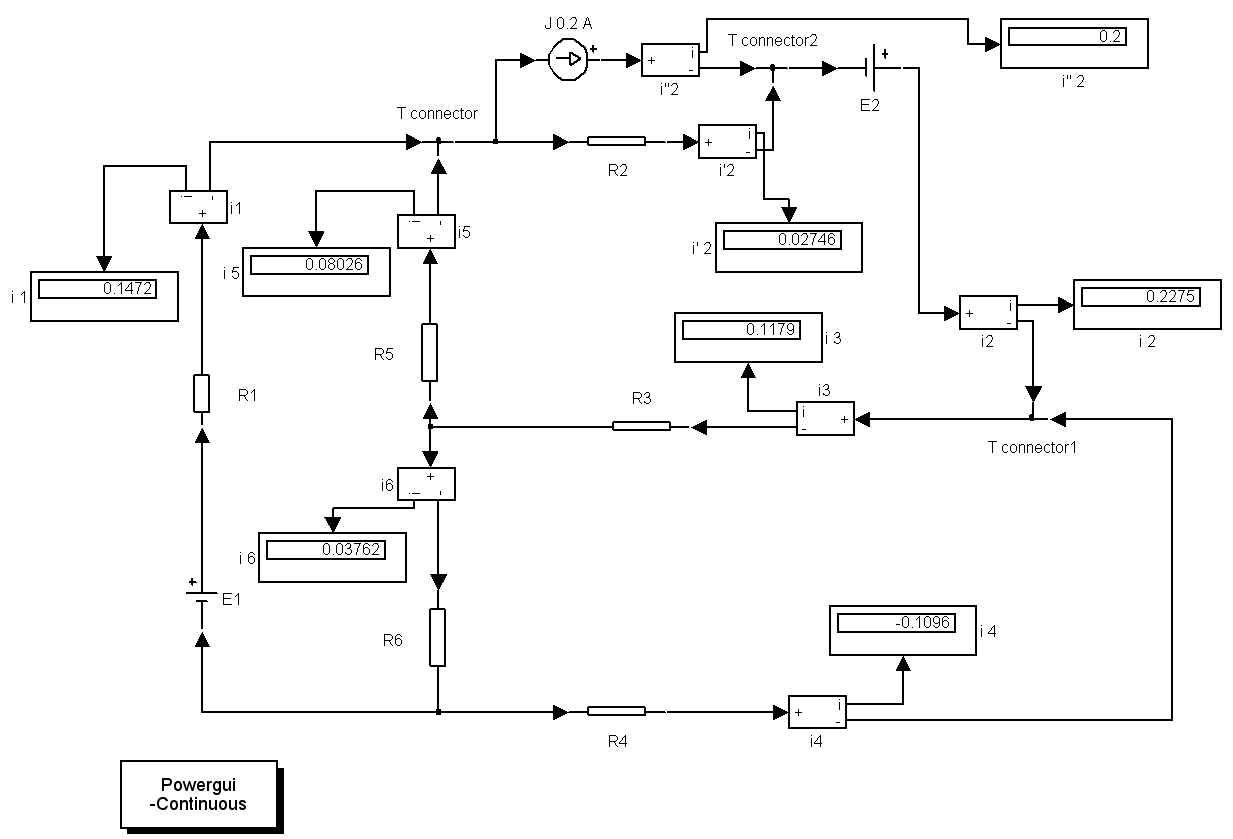


Рис 5.2

1.1. Задание к работе.

Набрать на компьютере и исследовать электрическую схему, представленную на рис. 5.1. Пример набранной схемы в Matlab представлен на рис 5.2.

1. 2. Порядок выполнения работы.

1. Набрать и исследовать модель в MATLAB (рис 5.2).
2. Промоделировать работу модели.

Таблица 5.1.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| R1 Ом | 220 | 560 | 160 | 360 | 260 | 270 | 280 | 660 | 360 | 460 | 560 | 660 | 760 |
| R2 Ом | 100 | 60 | 70 | 59 | 56 | 80 | 90 | 100 | 67 | 98 | 99 | 80 | 60 |
| R3 Ом | 140 | 100 | 100 | 90 | 80 | 60 | 120 | 120 | 40 | 120 | 90 | 110 | 120 |
| R4 Ом | 180 | 200 | 200 | 200 | 100 | 200 | 200 | 250 | 350 | 150 | 100 | 150 | 200 |
| R5 Ом | 160 | 250 | 270 | 220 | 300 | 160 | 220 | 300 | 170 | 180 | 190 | 200 | 220 |
| R6 Ом | 50 | 60 | 70 | 80 | 90 | 100 | 90 | 80 | 70 | 60 | 50 | 60 | 70 |
| E1 В | 30 | 24 | 26 | 30 | 32 | 34 | 36 | 38 | 40 | 42 | 44 | 46 | 48 |
| E2 В | 40 | 38 | 36 | 34 | 32 | 30 | 32 | 34 | 36 | 38 | 40 | 42 | 44 |
| J А | 0,4 | 0,2 | 0,6 | 0,2 | 0,4 | 0,2 | 0,5 | 0,2 | 0,8 | 0,2 | 0,4 | 0,2 | 0,5 |

1.3. Содержание отчета.

Отчет должен содержать:

* Заданную схему с параметрами.
* Схему модели, выполненную в MATLAB, с рассчитанными токами и напряжениями.
* Выводы.

1.4. Контрольные вопросы

1. Как запустить Simulink?
2. В какой библиотеке MATLAB находятся сопротивления?
3. В какой библиотеке MATLAB находятся измерительные приборы?
4. Как смоделировать источник постоянного тока?
5. Как проверить расчетное значение тока в сопротивлении R3?
6. Как измерить значение напряжения на сопротивлении R2?
7. Как измерить сопротивление между точками 1 и 2 схемы представленной на рис. 1.1?

# Лабораторная работа № 6. Расчет цепи переменного тока

*Цель работы: Расчет токов и напряжения в цепи переменного тока.*

6.1. Задание к работе.

В данной работе необходимо построить и рассчитать модель электрической цепи переменного тока в пакете Simulink (рис 6.1).

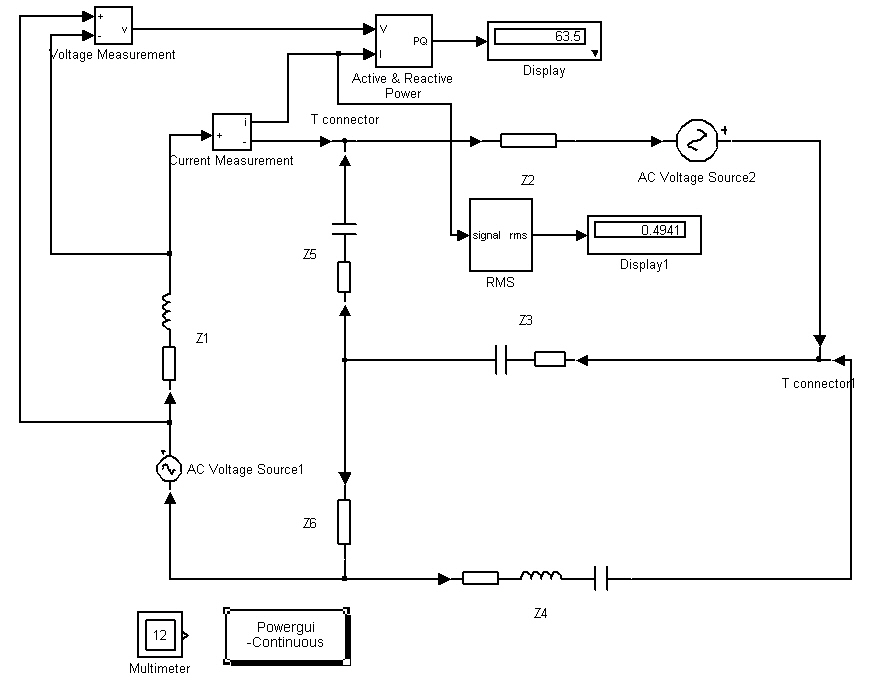


Рис 6.1

Варианты для расчета представлены в таблице 6.1.

Для получения результатов расчета необходимо использовать блоки «Мультиметр» и «Powerqui - Continuous». «Powerqui - Continuous» это графический интерфейс пользователя.

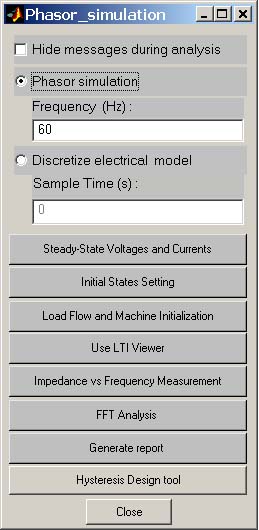
Пиктограмма:

p01

Назначение:

Блок является инструментом графического интерфейса пользователя и обеспечивает решение следующих задач:

* расчет схемы комплексным методом,
* расчет установившегося режима,
* дискретизация модели,
* задание начальных условий,
* инициализация трехфазных схем содержащих электрические машины, таким образом, чтобы расчет начался с установившегося режима,
* анализ схемы с помощью инструмента Simulink LTI-Viewer,
* определение полного сопротивление (импеданса) цепи,
* выполнение гармонического анализа,
* создание отчета,
* создание файла характеристик намагничивания для модели нелинейного трансформатора.
* Окно задания параметров:

Параметры блока:

«Hide messages during analysis» [Скрывать сообщения при проведении анализа]. Если флажок установлен, то подавляется вывод сообщений в командном окне MATLAB при выполнении расчетов.

«Phasor simulation» [Расчет схемы комплексным методом]. При установленном флажке выполняется расчет схемы комплексным методом. При этом необходимо задать частоту источников в графе Frequency.

«Frequency (Hz)» [Частота (Гц)]. Частота источников при расчете схемы комплексным методом. При других видах анализа параметр является недоступным.

«Discretize electrical model» [Выполнить дискретизацию модели]. При установленном флажке выполняется дискретизация модели. При этом необходимо задать шаг дискретизации в графе «Sample time.» «Sample time (s)» [Шаг дискретизации]. Параметр является доступным, если задан режим дискретизации модели.

При этом на пиктограмме блока будет показана величина этого параметра.

«Steady State Voltages and Currents» [Установившееся значения напряжений и токов]. Расчет установившихся значений переменных. При нажатии на кнопку открывается окно, в котором будут показаны соответствующие значения.

«Initial states Setting» [Установка начальных значений]. При нажатии на кнопку открывается окно, в котором отображаются начальные значения переменных. Эти значения можно изменять. Новые значения используются при расчете переходных процессов.

«Load Flow and Machine Initializations» [Инициализация схем содержащих электрические машины].

«Use LTI Viewer» [Использование LTI Viewer]. Применение инструмента Simulink LTI Viewer для анализа схемы.

«Impedance vs Frequency Measurements» [Определение импеданса цепи]

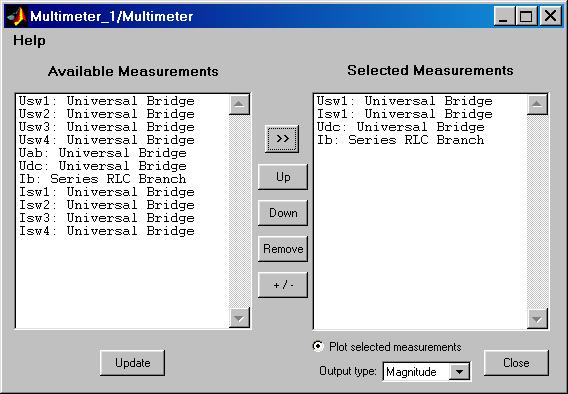
«FFT Analysis» [Гармонический анализ].

«Generate Report» [Создание отчета].

«Hysteresis Design Tool» [Инструмент расчета характеристики намагничивания].

Multimeter - Мультиметр

Его назначение - выполнять измерение токов и напряжений блоков библиотеки SimPowerSystem для которых в их окне диалога установлен параметр Measurements (измеряемые переменные). Окно задания параметров:



Параметры блока:

Available Measurements [Переменные, доступные для измерения]. В данной графе отображаются переменные (токи и напряжения) блоков схемы для которых в их окне диалога установлен параметр Measurements (измеряемые переменные). Обновление списка переменных можно выполнить с помощью клавиши Update.

Selected Measurements [Измеряемые переменные]. В данной графе указываются переменные, которые будут передаваться на выход блока Multimeter. Для управления списком измеряемых переменных можно использовать следующие клавиши:

* >> - Добавить выделенную переменную в список измеряемых.
* Up - Передвинуть вверх выделенную переменную в список измеряемых.
* Down - Передвинуть вниз выделенную переменную в список измеряемых.
* Remove -Удалить выделенную переменную из списка измеряемых.
* +/- - Изменить знак выделенной переменной.

Output signal: [Выходной сигнал]. Вид выходного сигнала блока. Выбор значения параметра возможен только, если с помощью блока Powergui установлен режим расчета на переменном токе (Phasor simulation). В этом случае значение параметра выбирается из списка:

* Magnitude - Амплитуда (скалярный сигнал).
* Complex - Комлексный сигнал.
* Real-Imag - Вектор, состоящий из двух элементов - действительная и мнимая составляющие сигнала.
* Magnitude-Angle - Вектор, состоящий из двух элементов - амплитуда и аргумент сигнала.

Блок может использоваться для измерения напряжений и токов вместо обычных измерителей - Current Measurement и Voltage Measurement. Список блоков, в окне параметров которых имеется графа Measurements, приведен в таблице 2.1.

Таблица 6.1

|  |  |
| --- | --- |
| № | Название блока |
| 1 | AC Current Source Parallel RLC Branch |
| 2 | AC Voltage Source Parallel RLC Load |
| 3 | Controlled Current Source PI Section Line |
| 4 | Controlled Voltage Source Saturable Transformer |
| 5 | DC Voltage Source Series RLC Branch |
| 6 | Breaker Series RLC Load |
| 7 | Distributed Parameter Line Surge Arrester |
| 8 | Linear Transformer Three-Phase Transformer (Two and Three Windings) |
| 9 | Mutual Inductance |
| 10 | Universal Bridge |

Выходным сигналом блока является вектор сигналов измеряемых переменных. На рис. 6.2 показана схема однофазного мостового выпрямителя, работающего на активно-индуктивную нагрузку.

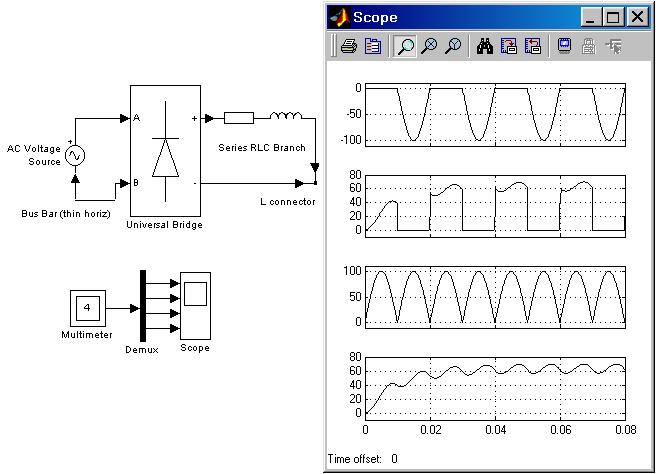


Рис. 6.2

С помощью блока Multimeter измеряются напряжение и ток одного из вентилей, напряжение на нагрузке и ток нагрузки.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вариант | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| R1 Ом | 220 | 560 | 160 | 360 | 260 | 270 | 280 | 660 | 360 | 460 | 560 | 660 |
| L1 H·10-6 | 10 | 20 | 30 | 40 | 10 | 15 | 10 | 30 | 50 | 20 | 40 | 30 |
| R2 Ом | 100 | 60 | 70 | 59 | 56 | 80 | 90 | 100 | 67 | 98 | 99 | 80 |
| R3 Ом | 140 | 100 | 100 | 90 | 80 | 60 | 120 | 120 | 40 | 120 | 90 | 110 |
| C3 F·10-6 | 10 | 20 | 30 | 40 | 10 | 15 | 10 | 30 | 50 | 20 | 40 | 30 |
| R4 Ом | 180 | 200 | 200 | 200 | 100 | 200 | 200 | 250 | 350 | 150 | 100 | 150 |
| L4 H·10-6 | 20 | 10 | 40 | 40 | 30 | 35 | 30 | 20 | 10 | 20 | 40 | 30 |
| R5 Ом | 160 | 250 | 270 | 220 | 300 | 160 | 220 | 300 | 170 | 180 | 190 | 200 |
| C5 F·10-6 | 40 | 50 | 30 | 60 | 70 | 65 | 50 | 30 | 70 | 40 | 30 | 20 |
| R6 Ом | 50 | 60 | 70 | 80 | 90 | 100 | 90 | 80 | 70 | 60 | 50 | 60 |
| E1 В | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 | 220 |
| E2 В | 127 | 127 | 127 | 127 | 127 | 127 | 127 | 127 | 127 | 127 | 127 | 127 |

6.2. Порядок выполнения работы

* Набрать и исследовать модель в MATLAB (рис 2.1).
* Промоделировать работу модели.

6.3. Содержание отчета. Отчет должен содержать:

* Заданную схему с параметрами.
* Схему модели, выполненную в MATLAB, с рассчитанными токами и напряжениями.
* Результаты расчета «SimPowerSystems Report».
* Выводы.

Пример.

Результаты расчета «SimPowerSystems Report».

SimPowerSystems Report.

generated by powergui,

16-Oct-2005 20:28:09

[1] Steady-State voltages and currents:

States at 50 Hz :

Il\_Z1 = 0.699 A 0.65░

Uc\_Z3 = 15.32 V -86.13░

Il\_Z4 = 0.4842 A -177.77░

Uc\_Z4 = 15.41 V 92.23░

Uc\_Z5 = 8.547 V -80.70░

Measurements at 50 Hz :

Voltage Measurement = 181.7 V 0.66░

Current Measurement = 0.699 A 0.65░

Ub: Z1 = 181.7 V 0.66░

Ub: Z2 = 193.1 V 3.05░

Ub: Z4 = 78.99 V 170.99░

Ub: Z5 = 59.69 V 1.07░

Ub: Z6 = 19.39 V -2.90░

Usrc: AC Voltage Source1 = 141.4 V 0.00░

Usrc: AC Voltage Source2 = 311.1 V 0.00░

Ib: Z1 = 0.699 A 0.65░

Ib: Z2 = 0.9653 A 3.05░

Ib: Z4 = 0.4842 A -177.77░

Ib: Z5 = 0.2685 A 9.30░

Ib: Z6 = 0.2154 A -2.90░

Sources at 50 Hz :

AC Voltage Source1 = 141.4 V 0.00░

AC Voltage Source2 = 311.1 V 0.00░

Nonlinear elements at 50 Hz :

[2] Initial values of States Variables:

Il\_Z1 = 0 A

Uc\_Z3 = 0 V

Il\_Z4 = 0 A

Uc\_Z4 = 0 V

Uc\_Z5 = 0 V

6.4. Контрольные вопросы

1. В чем состоит назначение блока «Powerqui - Continuous»?
2. Как задаются в модели значения индуктивности и емкости?
3. Какое значение тока измеряется с помощью блока «RMS»?
4. Что такое активная мощность?
5. Как вычислить активную мощность потребляемую схемой представленной на рис. 2.1?
6. Что такое реактивная мощность?
7. Как вычислить реактивную мощность потребляемую схемой представленной на рис. 2.1?
8. Как провести визуализацию значений физических величин измеряемых мультиметром?

# Лабораторная работа № 7. Создание и использование М-файлов

*Цель работы: изучение методики создания и использования М-файлов в MATLAB.*

7.1. М-файлы

Для более сложных задач число команд возрастает, и работа в командной строке становится непродуктивной. Использование истории команд, сохранение переменных рабочей среды или ведение дневника при помощи diary незначительно повышают производительность работы. Эффективное решение состоит в оформлении собственных алгоритмов в виде программ (М-файлов), которые можно запустить из рабочей среды или из редактора. Встроенный в MATLAB редактор М-файлов позволяет не только набирать текст программы и запускать ее целиком или частями, но и отлаживать алгоритм. Подробная классификация М-файлов приведена ниже.

7.1.1. Работа в редакторе М-файлов

Раскройте меню File рабочей среды MATLAB и в пункте New выберите подпункт M-file или нажмите кнопку New M-file на панели инструментов рабочей среды. Новый файл открывается в окне редактора М-файлов, которое приведено на рис. 7.1.

Наберите в редакторе команды для построения двух графиков на разных осях в одном графическом окне. Не обязательно набирать много команд — наша цель сейчас состоит в том, чтобы научиться выполнять команды из редактора М-файлов. Ограничьтесь командами, приведенными в следующем листинге:

x = 0:0.1:7;

f = exp(-x);

subplot(1, 2, 1)

plot(x, f)

g = sin(x);

subplot(1, 2, 2)

plot(x, g)

Сохраните теперь файл. Для запуска на выполнение всех команд, содержащихся в файле, следует выбрать пункт Run в меню Debug, или просто нажать «F5». На экране появится физическое окно Figure 1, содержащее графики функций. Результат эквивалентен последовательному выполнению команд листинга в командном окне. Однако если нужно построить график косинуса вместо синуса, достаточно просто изменить оператор присваивания g = sin(x) на g = cos (х) и запустить из редактора все команды. Аналогичные действия из командной строки потребовали бы больше времени.

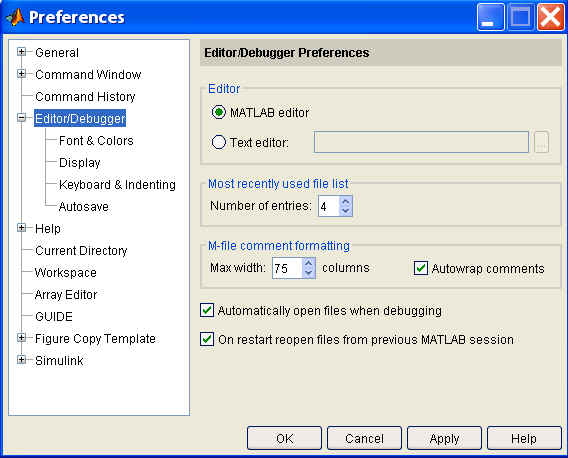
Очень удобной возможностью редактора М-файлов является выполнение части команд. Закройте графическое окно Figure 1. Выделите при помощи мыши, удерживая левую кнопку, первые четыре команды листинга. Затем откройте контекстное меню правой кнопкой мыши и выберете пункт «Evaluate Selection».

Если в М-файле при наборе сделана ошибка, то она выявляется в процессе исполнения. MATLAB выполняет команды до неправильно введенной, после чего в командное окно выводится сообщение об ошибке.

8.1.2. Настройки редактора М-файлов.

Для изменения настроек редактора М-файлов следует выбрать в меню «File» редактора или рабочей среды пункт «Preferences». Появляется одноименное диалоговое окно для настройки ряда компонент рабочей среды MATLAB.

В левой части окна отображены названия компонент, часть которых представлена раскрывающимся списком (слева находится знак +), позволяющим перейти к требуемой группе свойств. При изменении опций той или иной компоненты следует выбрать ее в списке и перейти к элементам управления в правой части окна «Preferences». Среди компонент есть и редактор М-файлов — раскрывающийся список Editor/Debugger. Рассмотрим далее наиболее важные настройки редактора.



При выборе заголовка раскрывающегося списка Editor/Debugger в правой части окна отображаются общие настройки, связанные с редактированием файлов в MATLAB. Панель «Editor» позволяет использовать вместо стандартного редактора MATLAB любой другой текстовый редактор, скажем, Notepad (Блокнот). Причем создание нового М-файла будет осуществляться по-прежнему в редакторе М-файлов, а открытие файлов — в выбранном редакторе.

К общим настройкам относится также длина списка последних открытых файлов (Number of entries), который располагается в меню «File» редактора или рабочей среды. При запуске MATLAB возможно автоматическое открытие тех файлов, с которыми велась работа во время предыдущей сессии, если при завершении работы редактор не был отдельно закрыт. Для этого следует установить флаг «On restart reopen files from previous МАТLAB session».

Сделайте активным пункт Display в левой части окна. В правой части окна появятся средства для изменения режимов. Например, в разделе «General Dysplay Options» расположены два флага. Флаг «Show line numbers» установлен указывает на то, что в рабочей области выделена колонка для нумерации строк текста в файле, а флаг «Enable data tips in edit mode» сброшен. Его ycтановка позволяет вывести значение переменной рабочей среды на всплывающую подсказку при наведении на переменную курсора мыши в редакторе. Разумеется, соответствующие переменные должны существовать в рабочей среде, поэтому их просмотр имеет смысл после выполнениями файла.

При записи выражений, содержащих много скобок, очень полезным оказывается автоматический контроль за их парностью, который настраивается в пункте «Keyboard&Indenting». Контроль может производиться в процессе набора, для чего следует установить флаг «Match parentheses while typing». В раскрывающемся списке «Show match with» вы можете выбрать, как при наборе выражения редактор будет показывать парную скобку: «Underline» — подчеркиванием, «Highlight» — выделением фона символа или Balance — выделением фона двух парных скобок. Редактор распознает незакрытые скобки и информирует вас одним из способов, представленных в раскрывающимся списке «Show mismatch with»: «Beep»— звуковым сигналом, «Strikethrough» перечеркнутым символом или «None» — никак. Возможен также быстрый поиск парной скобки в уже набранном выражении при наведении на нее курсора или оповещение об отсутствии таковой. Для настройки этой опции установите флаг «Match parentheses on arrow key or mouse movement» и обратив раскрывающимся спискам, расположенным под ним. И т.д

Можно использовать редактор М-файлов и без запуска MATLAB. Для этого дважды щелкните по значку с М-файлом в окне с содержимым папки, в которой он хранится. Файл откроется в редакторе М-файлов. Однако при этом редактор является самостоятельным приложением. Файл можно только редактировать, но не выполнять. Разумеется, расширение m в Windows должно быть ассоциировано с приложением meditor.exe (редактором М-файлов), что выбирается либо при установке MATLAB, либо в свойствах папки в Windows.

8.1.3. Типы М-файлов.

М – файлы в MATLAB бывают двух типов: файл - программы (Script M-Files), содержащие последовательность команд, и файл - функции (Function M-Files), в которых описывается функции определяемые пользователем.

Рассмотренные выше файл-программы являются последовательностью команд MATLAB, они не имеют входных и выходных аргументов. Для решения вычислительных задач и написания собственных приложений в MATLAB часто требуется программировать файл-функции, которые производят необходимые действия с входными аргументами и возвращают результат в выходных аргументах. Число входных и выходных аргументов зависит от решаемой задачи — может быть только один входной и один выходной аргумент, несколько и тех и других, или только входные аргументы. Возможна ситуация, когда входные и выходные аргументы отсутствуют.

Пример.

function f = myfun(x)

f = exp(-x).\*sqrt((x.^2+1)./(x.^4+1));

Имя файл-функции не обязательно должно совпадать с именем файла, однако обращение к ней происходит по имени файла. Например, если в файле FF содержится функция с заголовком g = init(z), то ее следует вызывать так:

>>f=ff(-0.9), а вовсе не >>f=init(-0.9).

Каталог, в котором содержатся файл-функции, должен быть текущим, или путь к нему должен быть добавлен в пути поиска, иначе MATLAB просто не найдет функцию или вызовет вместо нее другую с тем же именем (если она находится в каталогах, доступных для поиска).

8.2. Задание к работе

Построить и рассчитать динамические характеристики для заданной функции.

8.3. Порядок выполнения работы

а) Необходимо создать, согласно выбранного варианта, m – файл, сохранить его и исполнить:

s=zpk('s');

b0=7;

b1=-3;

a0=-4;

a1=4;

a2=5;

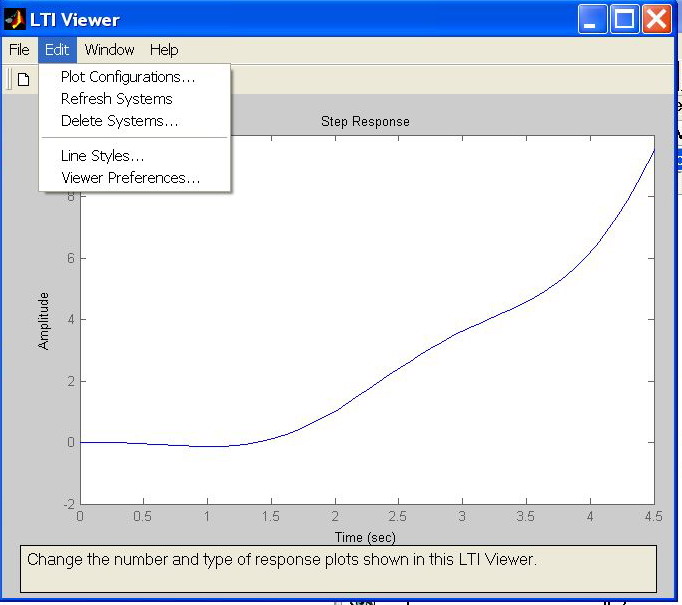
a3=0;

a4=1;

H=(-b1\*s+b0)/(s^4+0\*s^3+5\*s^2+4\*s-4);

ltiview(H)

в) В появившемся графическом редакторе необходимо открыть «Plot Configurations» и активировать окно с 6 графиками. Далее необходимо провести форматирование графиков, как показано на рис. 8.1.



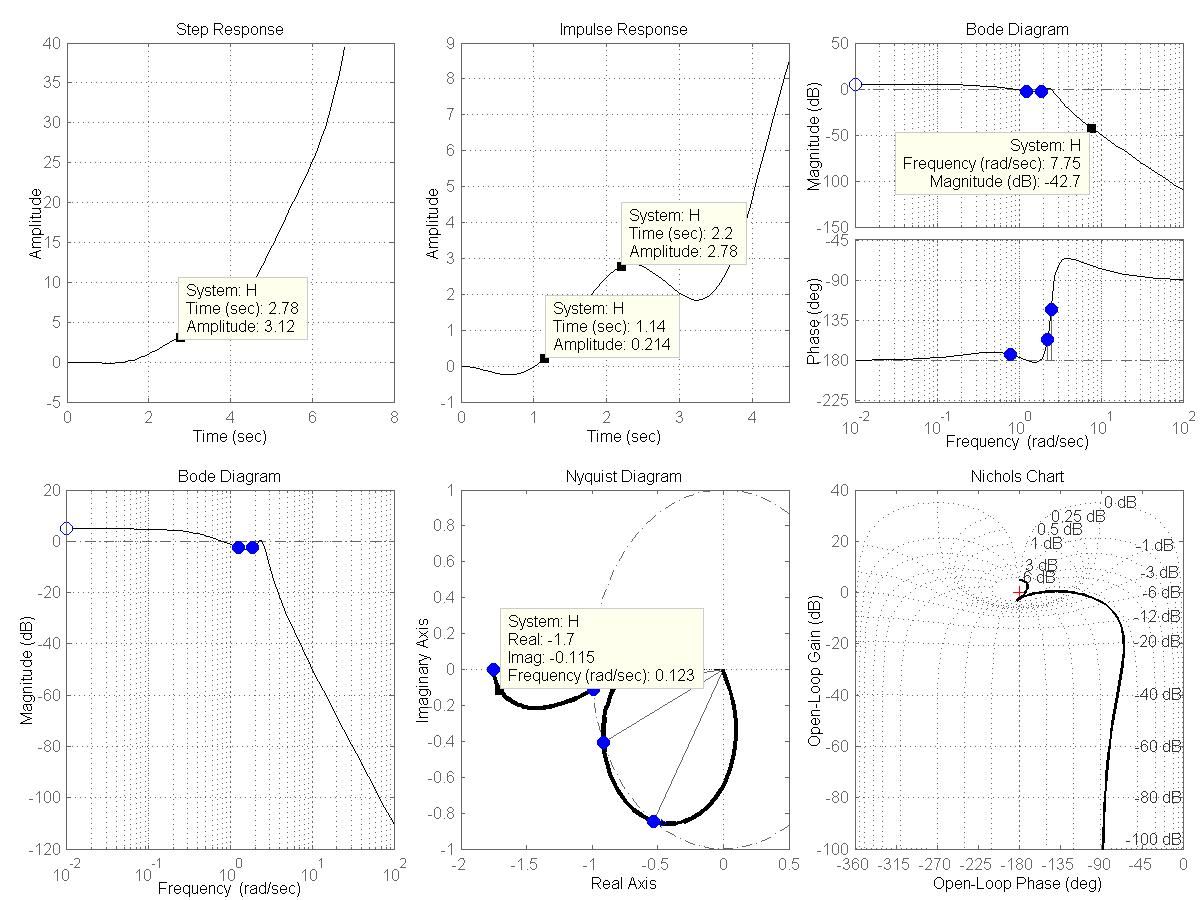


Рис.8.1

Таблица 8.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Вид передаточной функции | № | Коэффициенты полиномов | | | | | | |
|  |  | b0 | b1 | a0 | a1 | a2 | a3 | а4 |
| 1. | 0 | 3 | 1 | 2 | 3 | 0 | 1 |
| 2. | 2 | 6 | 4 | 0 | 1 | 5 | 1 |
| 3. | 0 | -3 | 5 | 2 | 0 | 2 | 1 |
| 4. | 4 | 2 | 3 | 4 | 5 | 3 | 1 |
| 5. | 0 | 1 | -2 | -2 | -3 | -2 | 0 |
| 6. | 0 | -3 | 2 | 4 | 2 | 3 | 9 |
| 7. | 8 | 0 | -3 | -4 | -6 | -4 | -1 |
| 8. | -4 | 6 | -2 | 5 | 5 | 0 | 1 |
| 9. | 6 | -8 | -7 | 0 | -6 | -3 | -1 |
| 10. | 2 | -1 | -3 | -1 | 0 | -7 | -2 |
| 11. | 0 | 2 | 8 | -3 | 7 | -7 | 1 |
| 12. | -5 | 0 | 3 | -8 | -2 | -1 | -6 |
| 13. | -7 | 1 | 2 | 0 | 5 | 2 | 9 |
| 14. | -6 | 4 | -4 | 1 | 0 | 6 | 3 |
| 15. | 2 | -2 | -1 | 5 | 3 | 0 | 9 |
| 16. | 7 | -6 | 0 | 5 | 8 | 2 | 2 |
| 17. | -2 | -8 | 2 | 0 | 4 | 3 | 3 |
| 18. | -7 | -1 | 6 | 9 | 0 | 4 | 2 |
| 19. | -3 | 7 | -4 | 4 | 5 | 0 | 1 |

**ЗАДАНИЕ**

1.Создать м-файл и простроить график

***Варианты заданий.***

|  |  |  |
| --- | --- | --- |
| № варианта | Функция | *х*0 |
| 1 |  | 5,5 |
| 2 |  | 2,75 |
| 3 |  | 3,1 |
| 4 |  | 4,21 |
| 5 |  | 6,32 |
| 6 |  | 4,75 |
| 7 |  | 2,35 |
| 8 |  | 8,29 |
| 9 |  | 4,56 |
| 10 |  | 1,23 |
| 11 |  | 7,55 |
| 12 |  | 3,64 |

2. Отделить корни трансцендентного уравнения графически.

*Варианты заданий.*

|  |  |  |  |
| --- | --- | --- | --- |
| № варианта | Задание | № варианта | Задание |
| 1 |  | 7 |  |
| 2 |  | 8 |  |
| 3 |  | 9 |  |
| 4 |  | 10 |  |
| 5 |  | 11 |  |
| 6 |  | 12 |  |

7.3. Содержание отчета. Отчет должен содержать:

* Разработанные программы.
* Результаты расчетов.
* Выводы.

7.4. Контрольные вопросы

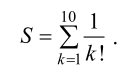
1. Чем отличается работа в командной строке и использование М-файла?
2. Как настроить ряд компонентов рабочей среды MATLAB?
3. Как настроить автоматический контроль скобок за их парностью?
4. Какие типы М-файла бывают?

# Лабораторная работа № 8. Программирование в MATLAB.

*Цель работы: Программирование* в MATLAB.

Рассмотрим некоторые программы.

8. 1. Разработаем файл-программу для вычисления суммы:



Алгоритм вычисления суммы использует накопление результата, то есть сначала сумма равна нулю, затем в переменную k заносится единица, вычисляется 1/k! (то есть 1/1!), добавляется к S и результат снова заносится в S. Далее k увеличивается на единицу, и процесс продолжается пока последним слагаемым не станет 1/10!.

Ниже приведенная Файл-программа вычисляет искомую сумму:

% ФАЙЛ-ПРОГРАММА ДЛЯ ВЫЧИСЛЕНИЯ СУММЫ

% 1/1! + 1/2! + … + 1/10!

% обнуление S для накопления суммы

S = 0;

% накопление суммы в цикле

for k = 1 : 10

S = S + 1 / factorial (k);

end

% вывод результата в командное окно

S

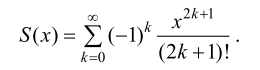
Если шаг цикла равен 1, то его можно не указывать.

Наберите файл-программу в редакторе М-файлов, сохраните в текущем каталоге в файле и выполните ее. Результат отображается в командном окне, т.к. в последней строке файл-программы содержится S без точки с запятой для вывода значения переменной S:

S = 1.7183

Обратите внимание, что остальные строки файл-программы, которые могли бы повлечь вывод на экран промежуточных значений, завершаются точкой с запятой для подавления вывода в командное окно.

Цикл for оказывается полезным при выполнении повторяющихся похожих действий в том случае, когда их число заранее определено. Обойти это ограничение позволяет более гибкий цикл while. Требуется найти сумму для заданного x (разложение в ряд sin(x)):



Конечно, до бесконечности суммировать не удастся, но можно накапливать сумму, пока слагаемые являются не слишком маленькими, скажем больше 10–10 (по модулю). Циклом **for** здесь не обойтись, т.к. заранее неизвестно значение k, обеспечивающее малость текущего слагаемого. Выход состоит в применении цикла **while**, который работает, пока выполняется условие цикла:

**while** условие цикла

команды MatLab

**end**

В данном примере условием цикла является то, что текущее слагаемое ! / k x k больше 10 –10 . Для записи условия в формате, понятной MatLab, следует использовать знак больше (>).

Текст файл-функции mysin, вычисляющей сумму ряда, выглядит так:

function s = mysin (x)

% Вычисление синуса разложением в ряд

% Использование: y = mysin (x), –pi<x<pi

s = 0;

k = 0;

while abs (x .^ (2\*k+1) / factorial (2\*k+1)) > 1.0e–10

s = s + (–1) ^k\*x. ^ (2\*k+1) / factorial (2\*k+1);

k = k + 1;

end

У цикла while, в отличие от for, нет переменной цикла, поэтому пришлось до начала цикла k присвоить ноль, а внутри цикла увеличивать k на единицу.

Сравним результат, построив графики функций mysin и sin на отрезке [−π π ] , на одних осях, например, при помощи fplot (команды можно задать из командной строки):

>> fplot (‘mysin’, [–pi, pi])

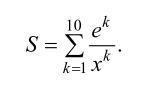
>> hold on

>> fplot (‘sin’, [–pi, pi], ‘k.’)

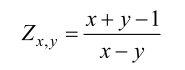
8. 2.Задания к работе

**Вариант 1**

1. Напишите файл-программу и файл-функцию для вычисления суммы

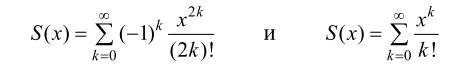


2. Напишите файл-программу для вычисления выражения



для x = 2, 3, …, n + 1 и y = 1, 2, …, n.

3. Напишите файл-функции для нахождения суммы ряда для заданного x (разложение в ряд cos(x) и ex ):

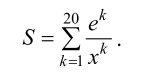


соответственно, пока слагаемые суммы больше 10-10 .

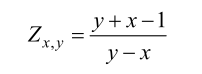
Сравните полученные результаты, построив графики функций mycos и cos на отрезке [−π π] , на одних осях, и myexp и exp, на отрезке [−2π 2π]

**Вариант 2**

1. Напишите файл-программу и файл-функцию для вычисления суммы

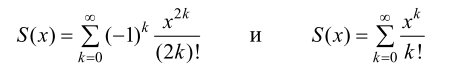


2. Напишите файл-программу для вычисления выражения



для x = 1, 2, …, n и y = 2, 3, …, n + 1.

3. Напишите файл-функции для нахождения суммы ряда для заданного x (разложение в ряд cos(x) и ex):



соответственно, пока слагаемые суммы больше 10-10.

Сравните полученные результаты, построив графики функций mycos

и cos на отрезке [−π π] , на одних осях, и myexp и exp, на отрезке [−2π 2π ].

8.3. Содержание отчета.

Отчет должен содержать:

* данные для программирования;
* m – файлы;
* результаты расчетов
* Выводы.

8.4. Контрольные вопросы

1. Какой язык программирования используется в MATLAB?
2. Как запустить настройку редактора М-файлов?
3. Какие типы М – файлы бывают в MATLAB?
4. Что такое файл-функция?
5. Как реализовать цикл в М – файле?

# Литература

1. В.Дьяконов. MATLAB. Анализ, идентификация и моделирование систем. Специальный справочник./ В.Дьяконов, В.Круглов. СПб.: Питер. 2001. - 448 с. - ISBN: 5-318-00359-1.
2. В.Дьяконов. MATLAB 6.5 SP1/7.0 Simulink 5/6. в математике и моделировании./ В.Дьяконов. М.: СОЛОН-Пресс. 2005 - 575 с. - ISBN 5-98003-209-6.
3. Ануариев И.Е. MATLAB 7./ Ануариев И.Е., Смирнов А.Б., Смирнова Е.Н.. Санк-Петербург. «БХВ-Петербург». 2005. - 1080 с. - ISBN 5-94157-494-0.
4. Черных И.В.. "Simulink: Среда создания инженерных приложений./ Черных И.В.. М.: Диалог –МИФИ. 2004, - 496 с. - ISBN 5-86404-186-6.
5. Герман-Галкин С. Г. Компьютерное моделирование полупроводниковых систем в MATLAB 6.0: Учебное пособие./ Герман-Галкин С. Г. СПб.: КОРОНА. 2001. - 320 с.- ISBN 5-7931-0158-6.
6. Герман-Галкин С. Г. Линейные электрические цепи. Лабораторные работы. / Герман-Галкин С. Г. СПб.: Учитель и ученик, КОРОНА принт. 2002. - 320 с.- ISBN. 9785793104654.
7. Бессонов Л.А. Теоретические основы электротехники. Электрические цепи./ Бессонов Л.А. М.: «Высшая школа», 1996. – 638 с.
8. Электротехника / Под ред. Пантюшина В.С. - М.: Высш. шк., 1976.
9. Черных И.В. Simulink: Инструмент моделирования динамических систем. <http://matlab.exponenta.ru/simulink/book1/index.php>

**Оглавление**

[Введение 4](#_Toc515471743)

[Лабораторная работа № 1. Основы работы с MATLAB 5](#_Toc515471744)

[Лабораторная работа № 2. Операции с векторами и матрицами в системе MATLAB 12](#_Toc515471745)

[Лабораторная работа № 3. Решение систем уравнений в MATLAB 18](#_Toc515471746)

[Лабораторная работа № 4. Аппроксимация полиномами в среде MATLAB 21](#_Toc515471747)

[Лабораторная работа № 5 Расчет цепи постоянного тока 24](#_Toc515471748)

[Лабораторная работа № 6. Расчет цепи переменного тока 26](#_Toc515471749)

[Лабораторная работа № 7. Создание и использование М-файлов 33](#_Toc515471750)

[Лабораторная работа № 8. Программирование в MATLAB. 42](#_Toc515471751)

[Литература 45](#_Toc515471752)