

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра электроснабжения

Проректор по учебной работе
О.Т. Лукичина
«15» 2017 г.



**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ В
ЭЛЕКТРОЭНЕРГЕТИКЕ**

Методические указания по выполнению лабораторных
для работ студентов направления подготовки 13.03.02
«Электроэнергетика и электротехника»

Курск 2017

УДК 510

Составители: В.И. Бирюлин, А.Н. Горлов, И.В. Ворначева, Д.В. Куделина

Рецензент
Доктор технических наук, профессор кафедры
«Электроснабжение»
А.В. Филонович

Программное обеспечение в электроэнергетике : методические указания по выполнению лабораторных работ для студентов направления подготовки 13.03.02 «Электроэнергетика и электротехника» / Юго-зап. гос. ун-т; сост.: В.И. Бирюлин, А.Н. Горлов, И.В. Ворначева, Д.В. Куделина. – Курск, 2017. – 45 с. – Библиогр.: с. 19.

Методические указания содержат рекомендации по выполнению лабораторных работ по дисциплине «Программное обеспечение в электроэнергетике». Рекомендации могут использоваться при выполнении лабораторных работ, максимально облегчают подготовку к зачету.

Предназначены преподавателям и студентам направления подготовки 13.03.02 «Электроэнергетика и электротехника»

Текст печатается в авторской редакции

Подписано в печать *15.12.17* Формат 60x84 1/16.
Усл. печ.л.3,37. Уч.-изд.л.3,052. Тираж 100 экз. Заказ *3054* Бесплатно.
Юго-Западный государственный университет.

305040, г. Курск, ул. 50 лет Октября, 94

ЛАБОРАТОРНАЯ РАБОТА №1

Основы работы с базами данных в автоматизированных системах

Цель работы: ознакомиться с использованием баз данных в автоматизированных информационных системах ведения паспортов, каталогов, справочников о основных видов электрооборудования.

Краткие методические указания.

Автоматизированные информационные системы (АИС) предназначены для ввода, хранения и обработки информации. Весьма перспективно применение таких систем для построения каталогов и справочников электрооборудования в различных энергетических службах и организациях, так как при этом резко сокращается время поиска нужной информации и повышается удобство ее использования.

АИС могут быть построены на основе различных программных систем, но наиболее удобными для этой цели являются системы управления базами данных (СУБД). База данных (БД) представляет собой таблицу, хранящуюся в памяти компьютера. Столбцы БД называются полями, каждое поле хранит данные одного типа (числового, символьного, логического, календарных дат) и одного значения (например, тип трансформатора и т. д.). Строки БД называются записями и хранят информацию об одном каком-то объекте.

БД создается следующей командой СУБД - CREATE < имя БД >. Угловые скобки выделяют изменяемую обязательную часть команды, при

ее вводе они не задаются. На первом этапе команды создается структура – определяются количество полей, их наименование, типы хранящихся данных и размеры. Например, при создании БД для трансформаторов, поле, в котором будет храниться информация по напряжению обмотки ВН, называют UVN. Его целесообразно сделать числового типа, общий размер поля 5 разрядов, из них для целой части (могут вводиться трансформаторы с напряжением 115, 220, 330 кВ) и один для дробной (37,5 или 10,5 кВ), один разряд будет автоматически отведен для разделителя между целой и дробной частью числа. После создания структуры производится ввод данных.

Более удобна для ввода, редактирования и просмотра данных команда BROWSE, показывающая активную или открытую БД в окне, команда BROWSE выводит данные в табличном виде. Для ввода информации о новом объекте создается предварительно пустая запись при действии команды BROWSE нажатием клавиш Ctrl – N, в которую затем производится ввод. Просмотр информации осуществляется с предварительным поиском нужной записи в БД (в простейшем случае пролистываем на экране).

При больших объемах информации для сокращения времени поиска можно использовать ограничения на вывод или задать поиск по какому-то условию. Ограничения могут задаваться по любому из существующих полей. Например, из БД трансформаторов выведем в окно команды BROWSE только те трансформаторы, у которых напряжение стороны ВН больше или равно 35 кВ, для этого команда имеет следующий вид:

BROWSE FOR UVN > = 35.

Для полей других типов используются соответствующие условия. Поиск записей выполняется командой LOCATE, например, необходимо найти в БД трансформатор с UВН = 35 кВ:

LOCATE FOR UVN = 35.

Если таких трансформаторов в БД несколько, то будет найден первый, для которого это условие выполняется. Чтобы результат поиска был более однозначным, следует использовать сложные условия поиска. Например, производится поиск трансформатора по напряжениям стороны ВН 35 кВ и стороны НН 6,3 кВ:

LOCATE FOR UVN = 35. AND. UVN = 6,3.

Удаление данных производится в два этапа. На первом, запись, подлежащая удалению, маркируется командой:

DELETE.

На втором этапе, запись или записи с маркерами удаляются командой:

PACK.

Порядок выполнения работы.

1. Для указанного преподавателем вида электрооборудования установить объем данных, который будет вводиться в БД, для каждого из вида данных определить имя поля, тип хранящейся в нем информации, размеры.
2. Создать соответствующую БД.
3. Выполнить ввод информации.
4. Повторить несколько раз операции поиска записи в БД по простому и сложному запросам.

5. Произвести просмотр, редактирование и удаление данных.
6. Повторить пункты 1-5 для других видов электрооборудования, указанных преподавателем.

Содержание отчета.

1. Структуры заданных БД.
2. Примеры информации (записи), хранящиеся в БД.
3. Запросы поиска информации.
4. Выводы по работе.

Контрольные вопросы.

1. Что называется БД ?
2. Основные типы данных, используемые в БД.
3. Команды просмотра и редактирования.
4. Поиск информации.
5. Основные типы БД

ЛАБОРАТОРНАЯ РАБОТА № 2

Программирование АИС

Цель работы: получение навыков в программировании АИС.

Краткие методические указания.

АИС для ведения паспортов, каталогов, справочников состоят из программ и БД. Программное обеспечение АИС в большинстве случаев состоит из отдельных, но связанных между собой модулей. Каждый модуль может разрабатываться отдельно, что сокращает трудозатраты на составление и наладку программы.

Один из модулей АИС является главным, он запускается при обращении к программе. Обычно главный модуль выполняет следующие операции: производство необходимых установок (например, открытие БД), формирование и вывод на экран готового меню АИС, организация выхода из работы.

Головное меню АИС обеспечивает выбор пользователей того или иного действия, например, ввода информации, просмотра редактирования. Для этого за каждым пунктом меню закрепляется определенный программный модуль.

Для организации ввода информации можно использовать следующий примерный текст программного модуля.

* * модуль ввода	– комментарий
DEFINE WINDOW ОКНО;	– определение окна
FROM 0,0 TO 24,79	на экране
ACTIVATE WINDOW ОКНО	– открытие окна

SET COLOR TO W+/G, GR+/R	– установка цветов
CLEAR	– очистка экрана
APPEND BLANK	– добавление пустой записи
@ 1,10 SAY «ТИП» GET TIP	– вывод области ввода
.....	в поле типы оборудования
READ	– активизация ввода
DEACTIVATE WIND ORNO	– закрытие окна

Просмотр и редактирование информации выполняется с предварительным поиском. Этот поиск можно осуществлять несколькими способами, но наиболее удобный – создание меню поиска, элементами списка которого будут являться уже введенные записи. Для меню целесообразно использовать одно или несколько символьных полей, однозначно определяющий искомый объект. Например, для поиска трансформаторов можно использовать тип и название подстанции, на которой он установлен.

Найденная таким образом информация может быть введена на экран различными способами. Наибольшие возможности дает команда типа

@ X, Y SAT < сообщение > GET < имя поля >.

Команда выводит в строку X и столбец Y на экране некоторое сообщение и справа от него содержание соответствующего поля. Вывод информации осуществляется на экран (по умолчанию), если сделать установку командой

SET DEVICE TO PRINT,

то вся информация направляется на нечитающее устройство (игнорируется при этом GET – часть), пока не будет произведена переустановка направления вывода на экран

SET DEVICE TO SCREEN.

В конце модуля просмотра информации ставится команда WAIT, приостанавливающая действие программы для изучения информации пользователем. Пауза в действии программы будет продолжаться до нажатия любой клавиши клавиатуре. Модуль редактирования информации должен после команды @ ... SAY ... GET иметь команду READ, активизирующую ввод новых значений в поле редактирования базы.

В рассмотренных модулях, так же как в модуле ввода, целесообразно использование окон, закрывающих весь экран и сосредотачивающих внимание пользователя на только одной операции.

Порядок выполнения работы.

1. Для указанного преподавателем вида электрооборудования составить структурную схему АИС.
2. Разработать и отладить главный модуль системы.
3. разработать и отладить модуль ввода, просмотра, редактирования данных и других операций, указанных преподавателем.
4. Запустить и опробовать работу АИС.
5. Повторить пункты 1-4 для другого вида электрооборудования.

Содержание отчета.

1. Структурная схема АИС.
2. Алгоритм работы одного из разработанных модулей АИС.
3. Текст модуля из пункта 2.
4. Выводы по работе.

Контрольные вопросы.

1. Виды модульного построения программ.
2. Вызов модулей и передача данных.
3. Способы отладки программы.
4. Использование изобразительных средств.
5. Индексация БД

ЛАБОРАТОРНАЯ РАБОТА № 3

Программирование автоматизированного рабочего места инженера службы линий (подстанций)

Цель работы: Приобрести навыки в создании программного обеспечения автоматизированных рабочих мест.

Краткие методические указания

Автоматизированное рабочее место (АРМ) служит для решения различных

задач, возникающих в процессе эксплуатации энергетического оборудования. В зависимости от тех или иных факторов, состав ее функции АРМ могут быть различными, но как правило, в их состав входят информационно-справочные системы, содержащие паспорта, ведомости, протоколы испытаний; системы подготовки документации и другие.

Службы линий и подстанции, входящие в состав энергетических предприятий, обслуживают большое количество объектов, поэтому применение АРМ обеспечивает значительное повышение эффективности труда.

Рассмотрим основные принципы построения информационно-справочной системы, входящей в состав АРМ. Получение информации об каком-либо объекте производится через предварительный поиск. Поиск производится по следующей цепочке:

подразделения – предприятия РЭС – подстанции – объекты.

Таким образом, осуществляется последовательный поиск по четырем БД. Целесообразно для проведения поиска использовать построение меню, списком которого является информация из БД.

БД подразделений имеет название PODR, а название подразделений хранятся в поле NAZVAN, с учетом этого примерный текст программы поиска выглядит так:

```
USE PODR IN 1           – открытие БД подразделений
USE RES IN 2           – открытие БД предприятий РЭС
USE PODST IN 3        – открытие БД подстанций
DEFINE POPUP PODRAZD ;
FROM 5,20 TO 15,40    – описание меню
PROMPT FIELD NAZVAN   поиска подразделений
ON SELE POPUP PODRAZD DO POISK 2 WITH PROMPT ( )
ACTIVATE POPUP PODRAZD
```

Приведенный пример обеспечивает открытие трех БД в различных рабочих областях, формирование меню поиска на экране с координатами (строка, столбец) левого верхнего угла 5,20 и правого нижнего 15,40. Элементы списка меню – названия подразделений из поля NAZVAN. Для выбора из меню необходимо установить курсор при помощи клавиши на соответствующий пункт меню и нажать клавишу ENTER. Действие программы после нажатия клавиши ENTER определяется командой ON SELE POPUP PODRAZD DO POISK 2 WITH PROMPT (). В рассматриваемом примере происходит вызов программного модуля нижнего уровня POISK 2 с передачей в него содержимого поля БД.

Модуль POISK 2 также должен сформировать меню из предприятий РЭС, входящих в выбранное подразделение. Предварительно должен

быть произведен отбор для составления меню. Данный отбор может быть произведен несколькими способами, одним из которых является индексирование с использованием ограничивающего условия.

Рассмотрим индексирование для БД RES в модуле POISK 2.

PARAMETR NPODR – прием названия выбранного подразделения

```
INDEX ON NAZV TO RES 1 FOR NAZV = NPODR
```

Приведенная команда индексирования производит упорядочение информации по полю NAZV, содержащему название предприятий РЭС и для дальнейшей обработки будут использоваться только те записи предприятий РЭС, входящих в подразделение, выбранное из первого меню. Таким образом, пользователь осуществляет последовательный выбор и в конце операции поиска находит нужный объект.

Объект в свою очередь может иметь большое число оборудования, поэтому интересующее оборудование выбирается также из меню. Дальнейшие действия с информацией производятся также как и в выше рассмотренных примерах.

Одним из направлений работы АРМ является ведение журналов ремонтов, отключений и т.п. Журнал содержит произвольное, заранее неизвестное количество информации, поэтому для его ведения удобно использовать поле типа Мемо. Информация вводится и просматривается при помощи специальных команд.

Порядок выполнения работы.

1. Изучить структуру БД, по которым будет происходить поиск.
2. Создать программные модули, обеспечивающие поиск объектов.
3. Создать программный модуль вывода информации об выбранном объекте на экран.
4. Запустить и отладить разработанные модули.
5. Произвести проверку работы всей системы в целом.

Содержание отчета.

1. Структуры БД.
2. тексты программных модулей.
3. Выводы по работе.

Контрольные вопросы.

1. Локализация информации в БД.
2. Поиск информации в БД.
3. Индексирование БД с компактным файлом.
4. Индексирование БД с мультииндексным файлом.
5. Ускоренный поиск информации в БД.

ЛАБОРАТОРНАЯ РАБОТА № 4

Автоматизированное рабочее место инженера службы ремонта

Цель работы: Изучить основы программирования расчетов графиков ремонта и других операций службы ремонта.

Краткие методические указания.

АРМ инженера службы ремонта, предназначенного для выполнения различных функций, к которым можно отнести: решение справочных ремонтных задач; частичной оценки технического состояния оборудования подстанций; автоматического формирования графиков испытаний, ремонтов, документации на их проведение.

Формирование графика ремонтов без применения средств вычислительной техники является весьма трудоемким процессом. Поэтому применение компьютеров здесь позволяет существенно сократить трудозатраты и время разработки графиков ремонтов.

Автоматизированное формирование графиков ремонтов производится по следующему алгоритму: база данных оборудования должна содержать информацию о дате предыдущего ремонта и межремонтном сроке. Для упрощения процесса считаем, что ремонты проходят через один и тот же срок. Кроме названной информации, необходимо иметь значения трудозатрат в человеко-часах или других единицах. При проведении более сложных ремонтов данная информация также усложняется (например, оборудование должно проходить через различные виды ремонта), в этом случае будет целесообразным использование нескольких БД.

Для рассмотрения примера такой программы предположим, что основная БД – VYKL (по выключателям) должна иметь кроме других полей следующие:

TIP – тип выключателя (C);

LAST_REM – дата последнего ремонта (D);

SROK_REM – межремонтный срок в годах (N).

Целесообразно использовать для работы две БД, основную и вспомогательную WSP, в которой будут храниться результаты расчета графика. С учетом всего этого примерный текст программы будет выглядеть так:

USE VYKL IN 1 – открытие основной БД;

USE WSP IN 2 – открытие вспомогательной БД;

SELE 2 – переход во вторую рабочую область.

ZAP – удаление открытых записей;

SELE 1 – переход в первую рабочую область;

GOD = YEAR (DATE ()) – получение значения года;

SCAN – начало цикла просмотра основной БД;

IF YEAR (LAST_REM) + SROK_REM = GOD – если ремонт в этом году;

SELE 2 – переход во вторую рабочую область;

APPEND VKFNK – добавление пустой записи;

REPL DAT_REM WITH A. LAST_REM + A. SROK_REM * 365,24

REPL TIP WITH A. TIP

REPL TRUD WITH A. TRUDOZAT

SELE 1

END SCAN – конец цикла просмотра БД.

После окончания работы программы во вспомогательной базе будет содержаться информация по всему оборудованию, которое должно быть отремонтировано в этом году. Полученный таким образом график выводится на экран или на печать. Он изучается, в него вносятся необходимые коррективы (работы более или менее равномерно распределяются по месяцам года), а окончательно составленный график выводится на печать. При вводе графика на печать используется предварительно запрограммированная форма произвольного вида, например, модуль печати может начинаться так:

SET DEVICE TO PRINT – направление вывода на принтер;

@ 1,5 SAY «_____»

@ 2,5 SAY «I»

@ 2,6 SAY «Тип выключателя»

@ 2,18 SAY «I»

.....

Определение суммарных трудозатрат наиболее просто сделать следующей командой:

CALC SUM (TRUD) TO S 1

Команда производит подсчет суммы по полю TRUD, содержащему значения трудозатрат и запоминает результат в переменной S 1. Значение суммы из переменной S 1 может быть выведено в итоговой строке распечатки графика.

Порядок выполнения работы.

1. Для заданной преподавателем БД создать вспомогательную БД.
2. Разработать и проверить в действии модуль расчета графика ремонта.
3. Разработать и проверить в действии модуль печати графика ремонта.
4. Разработать один из модулей: ввода результатов ремонта или переноса информации о проведенных ремонтах в основную БД из вспомогательной.
5. Повторить пункты 1-4 для другой БД.

Содержание отчета

1. Структура основной и вспомогательной БД.
2. Тексты разработанных программных модулей.
3. Выводы по работе.

Контрольные вопросы

1. Функции и команды, используемые при расчете графиков ремонтов.
2. Организация работы с несколькими БД.
3. Способы переноса информации между открытыми БД.
4. Обеспечение сохранности информации.
5. Резервное копирование БД

ЛАБОРАТОРНАЯ РАБОТА № 5

Основы работы с системой компьютерной математики SciLab

Цель работы: ознакомиться с основами работы в системе SciLab и приобрести первичные навыки работы в этой системе

Краткие методические указания.

Интенсификация научных исследований и инженерных разработок в условиях рыночных отношений, когда реализаций новых проектов ведется в жестких временных рамках, обусловила необходимость в программном обеспечении, позволяющем получать результат сложных математических задач в приемлемые сроки.

В ответ на эту потребность различными фирмами и институтами был создан ряд программных продуктов, к их числу относится Scilab. Scilab в отличие от аналогичных программных систем распространяется свободно.

Все приложения этого класса предоставляют пользователю командный интерфейс, с помощью которого производится описание математической формулировки задачи. После выполнения каждой команды можно видеть результат ее выполнения. Последовательным выполнением команд, по существу, создается программа, которая состоит из обращений к соответствующим высокоуровневым функциям, предоставляемым системой. Например, чтобы нарисовать график зависимости $f(x)$, необходимо вызвать функцию рисования графиков, сообщить этой функции вид $f(x)$ и, возможно, ее область определения.

Scilab - это система (язык) высокого уровня для выполнения мате-

математических расчетов. Система SciLab выполняет множество задач для поддержки научных инженерных работ, начиная от сбора и анализа данных до разработки приложений. Среда Sci Lab объединяет математические вычисления, визуализацию и мощный технический язык.

SciLab может использоваться в интерактивном и программном режимах. Интерактивный режим, характерен тем, что пользователь последовательно, шаг за шагом, решает задачу, используя SciLab как мощнейший калькулятор. В программном режиме разрабатывается и отлаживается программа, которая затем запускается на выполнение.

Основное окно Scilab содержит область просмотра и командную строку. В области просмотра отображаются команды текущего сеанса и результаты их выполнения. Командная строка — это область, предназначенная для ввода команд или текстовых комментариев. Нажатие клавиши «Enter» приводит к выполнению введенной в командной строке команды. Все символы, содержащиеся в строке программа воспримет как текст команды или названия переменных, но, если перед символами в строке ввода поставить знак «//», то запись не будет воспринята как команда, текст записи останется в области просмотра, а на следующей строке вновь появится приглашение «-->». Символ «//» позволяет делать текстовые комментарии как в рабочей области, так и в тексте программ.

В рабочей области Scilab можно определять переменные для дальнейшего их использования в различных выражениях. Определить переменную значит присвоить ей какое-либо значение. Оператором присваивания в Scilab является «=». Процедура присваивания оформляется следующим образом: имя переменной=значение переменной. Каждая пере-

менная перед использованием в выражении должна быть определена. В противном случае Scilab выдаст предупреждение об ошибке.

Переменным могут быть присвоены не только численные значения, например переменная может быть определена как строка символов. При определении переменной ей может быть присвоено значение результата вычисления. Тогда в результате выполнения команды на экран будет выведено:

имя переменной=результат вычисления.

Имя переменной не должно совпадать с именами встроенных процедур, функций и встроенных переменных системы и может содержать до 24 символов. Система различает большие и малые буквы в именах переменных. Т.е. ABC, abc, Abc, aBc это имена разных переменных. Выражение в правой части оператора присваивания может быть числом, арифметическим выражением, строкой символов или символьным выражением. Если речь идет о символьной или строковой переменной, то выражение в правой части оператора присваивания следует брать в одинарные кавычки.

Scilab позволяет выполнять инженерные и научные вычисления, такие как:

- решение нелинейных уравнений и систем;
- решение задач линейной алгебры;
- решение задач оптимизации;
- дифференцирование и интегрирование;
- обработка экспериментальных данных (интерполяция и аппроксимация, метод наименьших квадратов);

- решение обыкновенных дифференциальных уравнений и систем.

Для выполнения простейших арифметических операций Scilab использует следующие операторы:

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

Чтобы вычислить значение арифметического выражения, необходимо ввести его в командную строку и нажать «Enter». Результат вычисления появится в рабочей области.

Все функции, используемые в Scilab, можно разделить на два класса:

- встроенные;
- определенные пользователем.

В общем виде обращение к функции в Scilab имеет вид:

имя_переменной = имя_функции(переменная1 [, переменная2, ...])

где имя_переменной - переменная, в которую будут записаны результаты работы функции; этот параметр может отсутствовать, тогда значение, вычисленное функцией, будет присвоено системной переменной ans;

имя_функции - имя встроенной или ранее созданной пользователем функции;

переменная1, переменная2, ... - список аргументов функции.

Пакет Scilab снабжен достаточным количеством всевозможных

встроенных функций. Элементарные математические функции приведены в табл1.

Таблица 1 Элементарные математические функции

Функция	Описание функции
<i>Тригонометрические</i>	
<code>sin(x)</code>	синус числа x
<code>cos(x)</code>	косинус числа x
<code>tan(x)</code>	тангенс числа x
<code>cotg(x)</code>	котангенс числа x
<code>asin(x)</code>	арксинус числа x
<code>acos(x)</code>	арккосинус числа x
<code>atan(x)</code>	арктангенс числа x
<i>Экспоненциальные</i>	
<code>exp(x)</code>	Экспонента числа x
<code>log(x)</code>	Натуральный логарифм числа x
<i>Другие</i>	
<code>sqrt(x)</code>	корень квадратный из числа x
<code>abs(x)</code>	модуль числа x
<code>log10(x)</code>	десятичный логарифм от числа x
<code>log2(x)</code>	логарифм по основанию два от числа x

Функция, как правило, предназначена для неоднократного использования, она имеет входные параметры и не выполняется без их предварительного задания. Рассмотрим несколько способов создания функций в Scilab.

Первый способ - это применение оператора `deff`, который в общем виде можно записать так:

```
deff('[имя1,...,имяN]                                     =
имя_функции(переменная_1,...,переменная_M)',
'имя1=выражение1;...;имяN=выражениеN')
```

где `имя1,...,имяN` - список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений,

имя_функции - имя с которым эта функция будет вызываться,
переменная_1,...,переменная_M - входные параметры.

Второй способ создания функции это применение конструкции вида:

```
function[имя1,...,имяN]=имя_функции(переменная_1,...,переменная_
M)
тело функции
endfunction
```

где имя1,...,имяN - список выходных параметров, то есть переменных, которым будет присвоен конечный результат вычислений; имя_функции - имя с которым эта функция будет вызываться, переменная_1, ...,переменная_M - входные параметры.

Все имена переменных внутри функции, а также имена из списка входных и выходных параметров воспринимаются системой как локальные, т.е. считаются определенными только внутри функции.

Вообще говоря, функции в Scilab играют роль подпрограмм. Поэтому целесообразно набирать их тексты в редакторе и сохранять в виде отдельных файлов. Причем имя файла должно обязательно совпадать с именем функции. Расширение файлам-функциям обычно присваивают sci или sce.

Обращение к функции осуществляется так же, как и к любой другой встроенной функции системы, т.е. из командной строки. Однако функции, хранящиеся в отдельных файлах, должны быть предварительно загружены в систему, например, при помощи оператора `exec(имя_файла)` или командой главного меню File – Exec. . . , что, в общем, одно и то же.

Для работы с множеством данных удобно использовать массивы.

Например, можно создать массив для хранения числовых или символьных данных. В этом случае вместо создания переменной для хранения каждого данного достаточно создать один массив, где каждому элементу будет присвоен порядковый номер.

Таким образом, массив - множественный тип данных, состоящий из фиксированного числа элементов. Как и любой другой переменной, массиву должно быть присвоено имя.

Переменную, представляющую собой просто список данных, называют одномерным массивом, или вектором. Для доступа к данным, хранящимся в определенном элементе массива, необходимо указать имя массива и порядковый номер этого элемента, называемый индексом. Если возникает необходимость хранения данных в виде таблиц, в формате строк и столбцов, то необходимо использовать двумерные массивы (матрицы).

Порядок выполнения работы

1. Произвести расчеты по заданию преподавателя.
2. Построить графики полученных решений.
3. Исследовать устойчивость системы по характеристическому уравнению, заданному преподавателем.
4. Повторить пункты 1-3.

Содержание отчета

1. Задания, полученные от преподавателя.
2. Полученные решения.
3. Графики.

4. Характеристическое уравнение системы.
5. Корни характеристического уравнения системы.

Контрольные вопросы

1. Основные математические функции системы SciLab.
2. Команды построения графиков.
3. Операции с полиномами.
4. Действия над матрицами и векторами.
5. Основные типы переменных в системе SciLab

ЛАБОРАТОРНАЯ РАБОТА № 6

Применение системы SciLab для расчетов электрических цепей

Цель работы: ознакомиться с основами работы в системе SciLab, приобрести навыки в расчетах электрических цепей и исследованиях устойчивости в системе Sci Lab.

Краткие методические указания.

Система очень хорошо подходит для решения электротехнических задач, в том числе и для расчета цепей переменного тока, так как он позволяет использовать в вычислениях действительные и комплексные числа. рассмотрим пример расчета цепи:

Произведем расчет для следующих исходных данных: $U=220\text{В}$ (частота 50Гц), $R_1=10\text{ Ом}$, $R_2=15\text{ Ом}$, $R_3=20\text{ Ом}$, $L_1=50\text{ мГн}$, $L_2=60\text{ мГн}$, $C_1=150\text{ мкФ}$.

Ввод исходных данных и расчет производится в следующей последовательности:

$W=50$; - ввод частоты

$U=220$;

$R_1=10$;

$L_1=0,05$;

$R_2=15$; - ввод параметров цепи

$L_2=0,06$;

$R_3=20$;

$C_1=150\text{e-}6$;

$X_1=W\cdot L_1$;

$$X2 = W \cdot L2;$$

$$X3 = 1 / (W \cdot C1) \cdot j$$

$$Z1 = R1 + X1 \cdot j;$$

$$Z2 = R2 + X2 \cdot j; \quad \text{- расчет сопротивлений цепи}$$

$$Z3 = R3 - X3 \cdot j;$$

$$Z = Z1 + Z2 \cdot Z3 / (Z2 + Z3); \quad \text{- расчет эквивалентного сопротивления}$$

$$i1 = U / Z; \quad \text{- расчет тока первой ветви}$$

$$U23 = U - i1 \cdot Z; \quad \text{- расчет напряжения на второй и третьей ветви}$$

$$i2 = U23 / Z2; \quad \text{- расчет тока второй ветви.}$$

$$i3 = U23; \quad \text{- расчет тока третьей ветви.}$$

Если в конце строки не стоит знак `:`, то результат действия будет выведен на экран. Для построения на экране векторной диаграммы токов, как векторов исходящих из начала координат и имеющих угол и длину, определяемые действительной и мнимой частью комплексных чисел, представляющих эти векторы.

`tok = [i1, i2, i3]` ; – создание массива токов

`compass (tok)` – вывод векторной диаграммы.

Аналогичным образом осуществляется расчет и более сложных цепей.

При изучении переходных процессов часто возникает необходимость в оценке устойчивости системы. Для решения этой задачи дифференциальное уравнение системы преобразуется в характеристическое. Затем создается массив, содержащий все коэффициенты характеристического уравнения в порядке убывания степени слева направо. После этого применяется функция `roots` (), вычисляющая вектор, элементы которого являются корнями заданного уравнения. Далее из полученного вектора

выделяется элемент с наибольшей действительной частью и по нему оцениваются устойчивость системы. Исследование устойчивости и построение графика переходного процесса показывается на следующем примере.

Порядок выполнения работы

1. Произвести расчет заданной преподавателем цепи переменного тока, построить векторную диаграмму токов.
2. Построить график напряжений трехфазной системы по параметрам, заданным преподавателем.
3. Исследовать устойчивость системы по характеристическому уравнению, заданному преподавателем.
4. Повторить пункты 1-3.

Содержание отчета

1. Схема исследованной цепи.
2. Полученные значения токов и напряжений.
3. Векторные диаграммы.
4. Характеристическое уравнение системы.
5. Корни характеристического уравнения системы.

Контрольные вопросы

1. Основные математические функции системы SciLab.
2. Команды построения графиков.
3. Операции с полиномами.
4. Действия над матрицами и векторами.
5. Нахождение собственных чисел матрицы.

ЛАБОРАТОРНАЯ РАБОТА № 7

Применение системы SciLab для расчетов установившихся и переходных режимов работы электрических сетей

Цель работы: изучить применение системы SciLab для расчетов установившихся и переходных режимов работы электрических сетей.

Краткие методические указания

Расчеты как установившихся, так и переходных режимов, имеют важное значение в проектировании, эксплуатации электрических сетей. Они выполняются на основе уравнений узлового режима. Преимущества, обеспечившие им широкое применение, состоят в простоте формирования, больших возможностях эффективной организации вычислительного процесса и в меньшем количестве переменных по сравнению с контурными уравнениями.

Линейные уравнения узловых напряжений (при задании нагрузок токами в узлах) для трехфазной сети переменного тока могут быть записаны в следующем виде:

$$Y \cdot \dot{U} = -\sqrt{3} \dot{I} \quad (1)$$

где Y – полная комплексная матрица узловых проводимостей,

\dot{U} - вектор-столбец напряжений узлов,

\dot{I} - вектор-столбец токов в узлах.

В результате отбрасывания одного уравнения системы (1) приобретает следующий вид:

$$Y' \cdot (\dot{U} - \dot{U}_6) = -\sqrt{3} \cdot I \quad (2)$$

где Y' - неполная матрица узловых проводимостей;

\dot{U}_6 - вектор-столбец напряжений базисного узла.

Приведем систему (2) к виду:

$$Y' \cdot \dot{U} = -\sqrt{3} I + Y' U_6 \quad (3)$$

В системе (3) неизвестные значения напряжений узлов \dot{U} , остальные величины известны. Решение системы представляет собой установившийся режим работы электрической сети. Расчет этого режима производится по следующему алгоритму.

Предположим, что схема имеет n узлов, m ветвей. Все токи узлов одинаковы по величине – 0,1 кА, напряжение базисного узла – 115 кВ, сопротивления ветвей схемы заданы в виде R и X .

$Y = \text{zeros}(n)$; - создается нулевая матрица

$Y1 = 1 / (R_1 + X_1 \cdot i)$; - проводимость первой ветви

.....

$Yn = 1 / (R_m + X_m \cdot i)$; - проводимость m -й ветви

$Y(1,1) = Y1$; если к узлу 1 подходит

..... только первая линия.

Матрица Y заполняется нулевыми элементами по известным правилам. Правильность сопротивления матрицы Y можно проверить нахождением определителя – $\text{del}(Y)$. У правильно составленной матрицы его величина должна быть либо равной нулю, либо быть очень малой величиной.

ной (порядка 10^{-20}). Далее из матрицы Y выделяется неполная или сокращенная матрица:

$$YS = Y(2:n, 2:n);$$

$tok = [0.1; 0.1; \dots 0.1]$; - составляется вектор токов

$U_b = [115; 115; \dots 115]$; - вектор напряжений базисного узла

Решаем систему уравнений

$$U = YS \setminus (-1.73 \cdot tok + YS \cdot U_b) .$$

Получаем таким образом напряжения в узлах схемы. Находим токи в ветвях по следующей формуле:

$$I = (U_n - U_k) \cdot Y / \sqrt{3} ;$$

где U_n – напряжение начала ветви,

U_k – напряжение конца ветви,

Y – проводимость ветви.

Токи КЗ рассчитываются с использованием матрицы сопротивлений Z . Она получается обращением матрицы Y .

$$Z = \text{inv}(Y) .$$

Ток трехфазного КЗ в узле с номером K находится так:

$$I_k = U(k-1) / (Z(k,k) \cdot 1.73) .$$

Содержание отчета

1. Схема исследуемой сети.
2. Величины сопротивлений линий.
3. Значения напряжений.
4. Значения токов КЗ.

Порядок выполнения работы

1. Для заданной преподавателем схемы рассчитать сопротивления схемы и проводимостей линий.
2. Произвести расчет напряжений в узлах схемы.
3. Рассчитать значения токов в ветвях схемы.
4. Найти значения токов КЗ в заданных преподавателем узлах схемы.

Контрольные вопросы.

1. Функции и операторы линейной алгебры.
2. Операции с разреженными матрицами.
3. Решение систем нелинейных уравнений.
4. Решение систем уравнений с разреженными матрицами.
5. Проверка результата обращения матрицы

ЛАБОРАТОРНАЯ РАБОТА № 8

Численное решение дифференциальных уравнений в системе SciLab

Цель работы: изучить применение системы SciLab для решения дифференциальных уравнений.

Краткие методические указания

Решение дифференциальных уравнений производится при помощи встроенных функций, реализующих различные методы численного интегрирования. Наиболее просто находится решение для обыкновенных дифференциальных уравнений с начальными условиями, т.е. задачи Коши.

Эта задача состоит в нахождении функции, являющейся решением дифференциального уравнения произвольного порядка следующего вида:

$$y^{(n)} = f(t, y, y', \dots, y^{(n-1)}),$$

и начальными условиями при $t = 0$:

$$y(t_0) = u_0, y'(t) = u_1, \dots, y^{(n-1)}(t_0) = u_{n-1}. \quad (6.2)$$

Решение такой задачи производится по следующему алгоритму:

1. Преобразование дифференциального уравнения к системе дифференциальных уравнений первого порядка, в левой части которой должны находиться производные.

2. Составление специальной функции (программного файла) для вычисления правой части рассматриваемой системы уравнений.

2. Вызов наиболее подходящей встроенной функции численного решения дифференциальных уравнений.

3. Вывод полученных результатов на график для большей наглядности найденного решения.

Разберем решение дифференциальных уравнений на двух примерах.

Первый пример – решение задачи о движении ротора синхронного генератора при исследовании динамической устойчивости. Исходная схема системы приведена ниже (рис.1). В ней генератор работает на систему бесконечной мощности через двухцепную линию. В некоторый момент времени происходит отключение одной из цепей, и генератор начинает работать через одноцепную линию (рис.2), что приводит к возникновению электромеханического переходного процесса и изменению во времени внутреннего угла генератора δ между ЭДС генератора и напряжением системы.

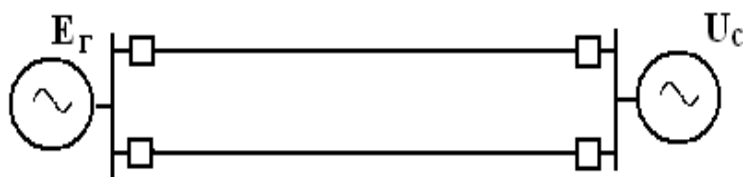


Рисунок 1 - Исходная схема рассматриваемой системы

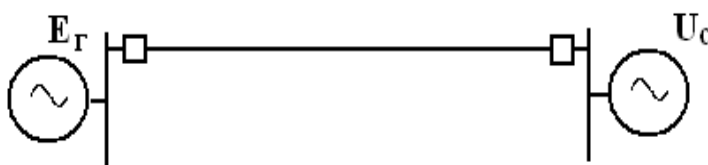


Рисунок 2 - Схема системы после отключений одной из цепей

Уравнение, описывающее движение ротора при изменении внутреннего угла генератора δ (без учета рассеяния энергии или демпфирования), имеет следующий вид в относительных единицах:

$$T_j \frac{d^2 \delta}{dt^2} = P_0 - P_A,$$

(1)

где T_j – постоянная инерции генератора;

P_0 – мощность приводного двигателя (турбины);

P_G – мощность генератора, определяемая как:

$$P_G = P_M * \sin \delta,$$

(2)

где P_M – максимум характеристики мощности генератора.

Найдем решение задачи для следующих начальных условий ($t = 0$): $\delta_0 = 35^\circ$; $P_0 = 1,0$; $P_M = 1,35$, $T_j = 8,18$.

Введем две вспомогательные функции y_1 и y_2 (количество вспомогательных функций равно порядку решаемого дифференциального уравнения). Они имеют следующие значения:

$$y_1 = \delta, \quad y_2 = \delta'.$$

(3)

Продифференцируем обе вспомогательные функции:

$$y_1' = \delta', \quad y_2' = \delta''.$$

(4)

Выразим y_2' из исходного уравнения (2):

$$y_2' = \frac{P_0 - P_M * \sin \delta}{T_j}.$$

(5)

Таким образом, исходное дифференциальное уравнение второго порядка преобразовано к форме Коши и приведено к системе из двух дифференциальных уравнений первого порядка следующего вида:

$$\begin{cases} y_1' = \delta', \\ y_2' = \frac{P_0 - P_M \cdot \sin \delta}{T_J}. \end{cases} \quad (6)$$

С учетом (4) система (6) приобретает вид:

$$\begin{cases} y_1' = y_2, \\ y_2' = \frac{P_0 - P_M \cdot \sin \delta}{T_J}. \end{cases}$$

Для решения полученной системы воспользуемся встроенной функцией `ode45`, реализующей метод Рунге-Кутты четвертого порядка. Входными аргументами функции `ode45` являются: имя файла-функции для вычисления правых частей, вектор с начальными и конечными значениями времени исследования процесса и вектор начальных условий. Выходные аргументы – вектор значений времени и матрица значений неизвестных (искомых) функций в соответствующие моменты времени. Эти вектор и матрица могут иметь большую размерность, поэтому результат решения лучше отображать на графике.

Составим файл-функцию с именем `rotor`. Этот программный файл состоит из двух строк. В первой строке объявляем функцию

```
function F = rotor(t,y)
```

Во второй строке записываем в виде вектора уравнения для вычисления y_1' и y_2' , используя для этого выражения (6.5) и (6.7).

```
F=[y(2); (1.0-1.35*sin(y(1)))*2*pi*50/8.18]
```

Коэффициент $2 \cdot \pi \cdot 50$ служит для перевода T_J из радиан в градусы.

Файл-функция будет иметь итоговый вид (вместе с комментарием).

```
%% rotor.m
```

```
%% файл-функция расчета правых частей для решения
```

```
%% уравнения движения ротора
```

```
function F = rotor(t,y)
```

```
F=[y(2); (1.0-1.35*sin(y(1)))*2*pi*50/8.18];
```

Файл программы решения уравнения движения ротора должен задать начальные условия по функциям y_1 и y_2 или их значения в начальный момент времени. Угол δ в начальный момент времени равен 35° , производная угла δ по времени равна 0, так как ротор вращается в момент времени, равный 0, с синхронной скоростью и угол δ не может измениться скачком из-за инерции ротора, что и обеспечивает равенство 0 производной $d\delta/dt$.

Далее необходимо определить вызов функции для решения исходного уравнения со следующими аргументами: имя файла-функции – 'rotor', вектор значений времени (от нуля до двух секунд) – [0 2.0], вектор начальных условий - y_0 . Прием полученных значений будет производиться в вектор T и матрицу Y, в первом столбце которой будут находиться значения угла δ в радианах.

Затем производится построение графика изменения угла δ от времени командой plot. Для этого используется вектор T и первый столбец матрицы Y, выделяемый как $Y(:,1)$, причем угол δ пересчитывается в градусы. На полученном графике наносятся надписи по обоим координатным осям и общий заголовок графика командами title('колебания ротора генератора'), xlabel('время в секундах'), ylabel('угол нагрузки в градусах').

Порядок выполнения работы

1. Составить дифференциальное уравнение для заданной преподавателем системы.
2. Составить файл-функцию для решения дифференциального уравнения по начальным условиям, заданным преподавателем.
3. Исследовать устойчивость системы по полученным результатам.
4. Повторить пункты 1-3.

Содержание отчета

1. Схема исследуемой системы.
2. Дифференциальное уравнение исследуемой системы.
3. Файл-функция решения дифференциального уравнения.
4. Решение дифференциального уравнения.
5. График решения дифференциального уравнения.

Контрольные вопросы

1. Основные математические функции системы SciLab.
2. Команды построения графиков.
3. Функции решения дифференциальных уравнений.
4. Выбор наиболее подходящей функции для решения дифференциальных уравнений.
5. Методы решения дифференциальных уравнений.

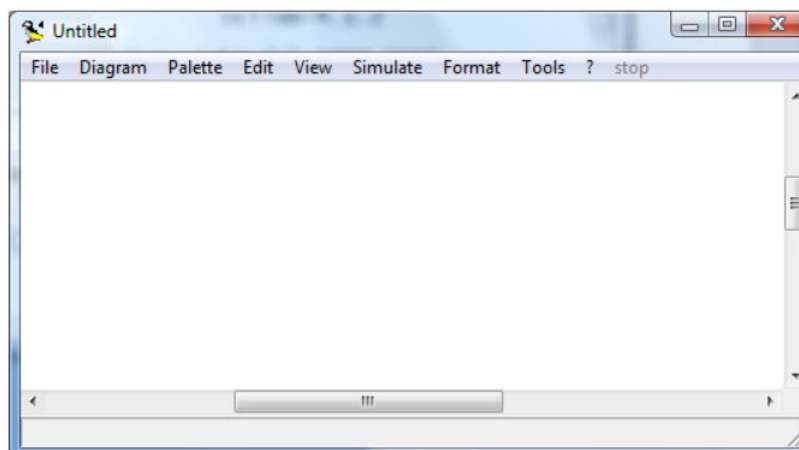
ЛАБОРАТОРНАЯ РАБОТА № 9

Использование пакета Scicos системы SciLab для исследования электрических цепей

Цель работы: изучить основы работы с пакетом Scicos, приобрести навыки в использовании пакета для исследования электрических цепей.

Краткие методические указания

В Scilab встроена система имитационного моделирования блочно организованных систем Scicos. Для вызова системы используется команда меню Scilab Applications=>Scicos. Генерируется пустое окно модели.



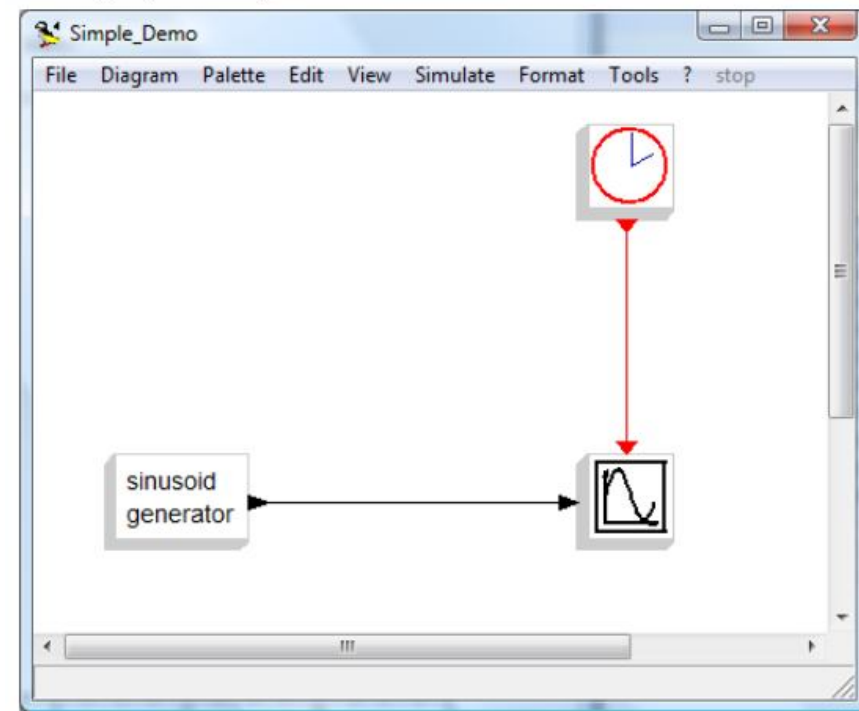
Главное меню окна модели содержит пункты:

<p>Edit View Simulate Format To</p> <ul style="list-style-type: none"> Undo Cut Copy Paste Duplicate Delete Move Smart Move Align Flip Rotate Left Rotate Right Add new block Block Documentation Label 	<p>Правка</p> <p>Отменить</p> <p>Вырезать в буфер</p> <p>Копировать в буфер</p> <p>Вставить из буфера</p> <p>Дубликат</p> <p>Удалить</p> <p>Перемещение</p> <p>Умное перемещение</p> <p>Выравнивание</p> <p>Перевернуть</p> <p>Вращение влево</p> <p>Вращение вправо</p> <p>Добавить новый блок</p> <p>Документация по блоку</p> <p>Метка</p>
<p>View Simulate Format Tools ? sto</p> <ul style="list-style-type: none"> Zoom in Zoom out Fit diagram to figure Default window parameters Available Parameters Grid 	<p><u>Вид</u></p> <p>Увеличить</p> <p>Отменить увеличение</p> <p>Зафиксировать диаграмму в рисунке</p> <p>Параметры окна по умолчанию</p> <p>Доступные параметры</p> <p>Сетка</p>
<p>Simulate Format Tools</p> <ul style="list-style-type: none"> Setup Compile Eval Debug Level Run 	<p><u>Моделирование</u></p> <p>Установка параметров</p> <p>Компиляция</p> <p>Вычисление</p> <p>Уровень отладки</p> <p>Запуск</p>

<p>Format Tools ? stop</p> <ul style="list-style-type: none"> Set grid Add color Default link colors Color Background color Show Block Shadow Resize Identification ID fonts Icon Icon Editor 	<p><u>Формат</u></p> <ul style="list-style-type: none"> Установить сетку Добавить цвет Цвета линий связи по умолчанию Цвет Цвет фона Показать тени у блока Изменить размер Идентификация Шрифт идентификатора Иконка Редактор иконки
<p>Tools ? stop</p> <ul style="list-style-type: none"> Create Mask Remove Mask Customize Mask Save Block GUI Get Info Details Browser Code Generation Shortcuts Activate Scilab Window Calc 	<p><u>Инструменты</u></p> <ul style="list-style-type: none"> Создать маску Удалить маску Сохранить блок ГИП Получить информацию Детали Браузер Генерация кода Активизировать окно Scilab Калькулятор
<p>? stop</p> <ul style="list-style-type: none"> Help Scicos Documentation Demos About Scicos 	<p><u>Справка</u></p> <ul style="list-style-type: none"> Вызов справки Документация по Scicos Демосы О Scicos
<p>Stop</p>	<p>Стоп</p>

Модель составляется из заранее запрограммированных блоков путем их переноса из библиотеки в окно модели. Доступ к библиотекам обеспечивается браузером, окно которого вызывается командой `Palette =>PalTree` из меню модели. В отдельном окне в виде дерева отображается палитра компонент. Альтернативный вариант - командой `Palette=>Palettes` из выпадающего списка вызывается подбиблиотека компонент.

Из перечня компонент нужно выбрать нужный и перенести его в окно модели нужный компонент. Затем в окне модели компоненты необходимо в визуальном режиме мышью соединить друг с другом. После заполнения блоками окно модели имеет вид, подобный этому:



В данной модели использован генератор синусоиды и виртуальный осциллоскоп. При запуске имитации в отдельном графическом окне отображается график синусоиды.

Порядок выполнения работы.

1. Собрать модель, состоящую из источника переменного напряжения, RLC – цепи и измерительного прибора Multimeter.
2. Установить параметры блоков по заданию преподавателя.
3. Запустить модель.
4. Проверить значения параметров.

5. Изменяя частоту источника питания, получить резонанс напряжений в исследуемой цепи.
6. Повторить пункты 1-5 для новых значений, заданных преподавателем.

Содержание отчета.

1. Блок-схема исследуемой модели.
2. Исходные параметры схемы.
3. Параметры, измеренные блоком Multimeter.
4. Зависимости тока в цепи от частоты.

Контрольные вопросы.

1. Выбор блоков из библиотек и установки их параметров.
2. Операции с блоками.
3. Измерение параметров работы блок-схемы.
4. Вывод зависимостей параметров от времени на экран.
5. Соединение блоков между собой.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Visual FoxPro 9.0. Наиболее полное руководство в подлиннике [Текст]: наиболее полное руководство в подлиннике/ Т.П. Агафонова, В.Б. Клепинин, - СПб.: БХВ-Петербург Год: 2008. - 1215 с.

2. Кузин А.В. Разработка баз данных в системе Microsoft Access [Текст]: учебник / А.В. Кузин, В.М. Демин – 3е изд. М.: ФОРУМ, 2012 – 224 с.

3. Андриевский А.Б. Решение инженерных задач в среде Scilab. [Электронный ресурс]: учебное пособие / А.Б. Андриевский, Б.Р. Андриевский, А.А. Капитонов, А.Л. Фрадков — СПб.: НИУ ИТМО, 2013. — 97 с.

4. Плещинская И. Е. Интерактивные системы Scilab, Matlab, Mathcad [Электронный ресурс]: учебное пособие / И. Е. Плещинская, А. Н. Титов, Е. Р. Бадертдинова, С. И. Дуев ; Министерство образования и науки России, ФГБОУ ВПО «Казанский национальный исследовательский технологический университет». - Казань : Издательство КНИТУ, 2014. - 195 с.