

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего
образования**

**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)**

**ЛАБОРАТОРНЫЕ И САМОСТОЯТЕЛЬНЫЕ РАБОТЫ ПО
ДИСЦИПЛИНЕ «ПАКЕТЫ ПРИКЛАДНЫХ ПРОГРАММ»**
*Методические указания для студентов направления «Экономика» и
«Менеджмент»*

Томск
2018

Методическое пособие для студентов ВУЗов посвящено изучению прикладных математических пакетов Smath Studio и Scilab. Описываются основные правила работы с математическими пакетами, особенности работы в Smath Studio и Scilab.

Составитель: Е.А. Шельмина

СОДЕРЖАНИЕ

Введение	4
Раздел 1. Лабораторные работы	4
Лабораторная работа №1	4
Лабораторная работа №2	7
Лабораторная работа №3	10
Лабораторная работа №4	11
Лабораторная работа №5	13
Лабораторная работа №6	17
Раздел 2. Самостоятельная работа	17
Список литературы.....	18

Введение

Этот курс предназначен для практического изучения прикладных математических пакетов Smath Studio и Scilab. В методических указаниях основное внимание уделяется принципиальным моментам, которые необходимы для успешного выполнения практических заданий.

Раздел 1. Лабораторные работы

Лабораторная работа №1

Структура окон Smath Studio и Scilab

Цель работы: изучение интерфейса пакетов Smath Studio и Scilab.

Структура окна Smath Studio

Программа Smath Studio предназначена для численного и аналитического решения математических задач (решения уравнений и систем, нахождения экстремумов функций, вычисления производных и интегралов, решения дифференциальных уравнений) [1].

Позволяет работать с формулами, текстами, графиками, а также выполнять программирование вычислительных процессов и производить расчеты как в численном, так и в аналитическом виде.

Программа состоит из 3 областей: основное меню, инструментальная панель, рабочее поле.

Основное меню состоит из основных команд для работы с документом в целом, такие как: вставить, вырезать, открыть, сохранить... а также содержит математический справочник и набор примеров [1].

Панель инструментов (ПИ) разделена по категориям:

а) панель «Арифметика» содержит цифры, математические символы, и основные операции:

– оператор присвоения « := » служит для присвоения переменным каких-либо значений, численных либо символьных;

– оператор численного вычисления « = » служит для получения численного результата, он применим как к выражениям, так и к переменным;

– оператор символьного вычисления « → » позволяет вычислять символьный результат;

б) панель «Матрицы» содержит команды для работы с матрицами. Позволяет находить определитель матрицы, транспонировать ее, находить минор, а также содержит команду векторного умножения, потому что векторы программа рассматривает как матрицу с одним столбцом (или строкой);

в) панель «Булева» содержит набор команд для булевой алгебры, а также позволяет задавать логические операции в командах ветвления и циклах;

г) панель «Функции» содержит набор часто используемых функций, таких как: \sin , \cos , \log и т. п., а также 2 кнопки «2d» и «3d», эти кнопки позволяют вставить соответственно 2-мерные и 3-мерные графики;

д) панель «График» позволяет вращать, перемещать, увеличивать/уменьшать графики функций;

е) панель «Программирование» содержит 4 функции программирования, таких как: ветвление «IF», цикл с предусловием «WHILE», цикл со счетчиком «FOR», вспомогательная функция «LINE»;

ж) последние две панели называются одинаково «Символы» и содержат греческие символы.

Рабочее поле занимает самую большую часть программы, здесь выполняются все расчеты. Основным элементом поля является курсор ввода, он выглядит как красный крестик.

Структура окна Scilab

Рассмотрим основные компоненты окна программы Scilab.

После запуска Scilab на экране появиться основное окно приложения. Окно содержит меню, панель инструментов и рабочую область. Признаком того, что система готова к выполнению команды, является наличие знака приглашения -->, после которого расположен активный (мигающий) курсор. Рабочую область со знаком приглашения обычно называют *командной строкой*. Ввод команд в Scilab осуществляется с клавиатуры. Нажатие клавиши *Enter* заставляет систему выполнить команду и вывести результат.

Понятно, что все выполняемые команды не могут одновременно находиться в поле зрения пользователя. Поэтому, просмотреть ту информацию, которая покинула видимую часть окна можно, если воспользоваться стандартными средствами просмотра, например, полосами прокрутки или клавишами перемещения курсора Page Up, Page Down.

Клавиши «Стрелка вверх» и «Стрелка вниз» так же управляют курсором, однако в Scilab они имеют другое назначение. Эти клавиши позволяют вернуть в командную строку ранее введенные команды или другую входную информацию, так как вся эта информация сохраняется в специальной области памяти. Так, если в пустой активной командной строке нажать клавишу «Стрелка вверх», то появится последняя вводимая команда, повторное нажатие вызовет предпоследнюю и так далее. Клавиша «Стрелка вниз» выводит команды в обратном порядке. Таким образом, можно сказать, что вся информация в рабочей области находится или в зоне просмотра или в зоне редактирования. Важно знать, что в зоне просмотра нельзя ничего исправить или ввести. Единственная допустимая операция, кроме просмотра, это выделение информации с помощью мыши и копирование ее в буфер обмена, например, для дальнейшего помещения в командную строку. Зона редактирования это фактически командная строка [2,3].

Кроме того, существуют особенности ввода команд. Если команда заканчивается точкой с запятой «;», то результат ее действия не отображается в командной строке. В противном случае, при отсутствии знака «;», результат действия команды сразу же выводится в рабочую область [3].

Текущий документ, отражающий работу пользователя с системой Scilab, содержащий строки ввода, вывода и сообщения об ошибках принято называть *сессией*. Значения всех переменных, вычисленные в течение текущей сессии, сохраняются в специально зарезервированной области памяти, называемой рабочим пространством системы. При желании, определения всех переменных и функций, входящих в текущую сессию, можно сохранить в виде файла, саму сессию сохранить нельзя.

Основные команды главного меню Scilab

Главное меню системы содержит команды предназначенные для работы с файлами, настройки среды, редактирования команд текущей сессии и получения справочной информации. Кроме того, с помощью главного меню можно создавать, редактировать, выполнять отладку и запускать на выполнение так называемые файлы-сценарии Scilab, а так же работать с графическими приложениям пакета.

Работа с файлами

Пункт меню File предназначен для работы с файлами. Рассмотрим назначение представленных в нем команд:

- New Scilab - открывает новое окно Scilab, фактически пакет запускается повторно;
- Exec... - запуск на выполнение созданной ранее Scilab-программы (файлы с расширением sce или sci);

- Open - открывает окно для загрузки созданного ранее файла, рисунка или модели;
- Load - открывает окно для загрузки файлов, информация в которых хранится в виде машинных кодов, при их открытии в память компьютера загружаются определенные ранее переменные и функции;
- Save - сохранение всех определенных в данной сессии переменных и функций в виде файла с расширением sav или bin;
- Change Directory - смена текущего каталога, выводит окно настройки путей файловой системы;
- Get Change Directory - выводит в командную строку имя текущего каталога;
- Print Setup... - выводит окно настройки параметров печати;
- Print - печать текущей сессии;
- Exit - выход из системы Scilab.

Редактирование команд текущей сессии

Пункт меню Edit содержит следующие команды:

- Select All - выделение всех команд текущей сессии;
- Copy - копирование выделенного объекта в буфер;
- Paste - вставка объекта из буфера;
- Empty Clipboard - очистка буфера обмена;
- History - группа команд предназначенных для редактирования командной строки.

Настройка среды

Команды настройки среды пакета представлены в меню Preferences:

- Language - предлагает выбрать из списка язык интерфейса (английский, французский);
- Colors - позволяет установить цвет шрифта (Text), цвет фона (Background) или цвета принятые по умолчанию (Default System Colors);
- Toolbar - выводит или удаляет панель инструментов;
- Files Association- предлагает установить типы поддерживаемых файлов;
- Choose Font - выполняет настройки шрифта (гарнитура, начертание, размер);
- Clear History -очищает рабочее пространство;
- Clear Command Window - очищает рабочее окно;
- Consol - активизирует консольное приложение.

Справочная система

Команда главного меню ? открывает доступ к справочной системе Scilab. В справочной системе информацию можно искать, воспользовавшись содержанием, в списке, упорядоченном по алфавиту, по ключевому слову или фразе.

С помощью команды Scilab Demos можно осуществить просмотр демонстрационных примеров.

Редактирование и отладка файлов-сценариев

Файл-сценарий - это список команд Scilab сохраненный на диске. Для подготовки, редактирования и отладки файлов-сценариев служит специальный редактор SciPad, который можно вызвать, выполнив команду главного меню Editor. В результате работы этой команды будет создан новый файл-сценарий. По умолчанию он имеет имя Untitled1.sce.

Окно редактора файлов-сценариев выглядит стандартно, то есть имеет заголовок, меню, панели инструментов, строку состояния. Ввод текста в окно редактора файла-сценария осуществляется по правилам принятым для команд Scilab [3].

Задание. Изучите пункты меню пакетов Smath Studio и Scilab. Изучите подпункт «Справочник» пункта «Помощь».

Лабораторная работа №2

Арифметические операции. Целые и рациональные числа, константы.

Синтаксис команд. Стандартные функции.

Цель работы: познакомится с основными арифметическими операциями пакетов Smath Studio и Scilab, научится выполнять простейшие математические вычисления.

Арифметические операции. Целые и рациональные числа. Константы. Синтаксис команд. Стандартные функции в Smath Studio

Для проведения каких-либо математических вычислений, в математическом пакете Smath Studio, нужно установить курсор на рабочем поле, ввести выражение, поставить знак "=" и нажать Enter. Линия подчёркивания указывает на те символы, к которым будет применено действие арифметического знака или функции. Изменить её положение можно с помощью клавиши «пробел» [3].

Определение переменной

Присваивание значения переменной осуществляется знаком «:=», который можно ввести из панели «Арифметика» или с клавиатуры.

Согласно синтаксису рассматриваемого пакета, введённая заранее переменная или функция должна быть записана левее или выше того выражения, в котором она используется. При этом её изменение приведёт к тому, что весь лист будет пересчитан (незаменимое свойство при организации многократных объёмных вычислений). Промежуточные расчёты можно скрыть из поля зрения с помощью инструмента «Область» (меню Вставка — Область).

Текстовый комментарий может быть введён в любой области рабочего окна, при этом никакие специальные знаки не требуются. Пользователь может менять цвет шрифта, фона, а также выделять выражение рамкой.

Программа знает многие математические и физические константы, умеет работать с размерностями. Весь лист с расчётами может быть сохранён и в дальнейшем открыт для просмотра и редактирования.

Для пакета Smath Studio есть определенные правила записи выражений:

1. используемая переменная или функция должна быть объявлена левее или выше того выражения, где она используется в вычислении;
2. если переменная переобъявлялась, то будет использовано то значение, которое встретилось самым последним перед использованием в вычислениях;
3. при объявлении переменной в выражении можно использовать встроенные и ранее объявленные функции, ранее объявленные переменные и их сочетания. Если используемые в выражении переменные ранее не объявлялись, то результат можно будет получить только в символьном виде;
4. переменная не обязательно должна вычисляться как числовое значение, допускается присваивать имена выражениям, дающим при вычислении матрицу;
5. для символьных вычислений объявлять переменные заранее не требуется, если не нужно, чтобы при преобразовании выражений были подставлены их значения.

При работе с вещественными числами иногда требуется настраивать точность ответа (количество знаков после запятой). Для этого надо выполнить команду: Сервис - Опции - вкладка «Вычисление» - «Точность ответа». Дробная часть в числах отделяется запятой.

Рассмотрим подробно синтаксис различных команд в Smath Studio.

Правила ввода текста на рабочем листе Smath Studio

В SMath Studio есть возможность вставлять текстовые области на рабочем листе. Например, Вы можете сделать примечания о проделанных вычислениях. Для того, чтобы ввести текст, необходимо [1]:

1. Щелкнуть в выбранном месте рабочего листа.

2. Выбрать Текстовая область из меню Вставка, или нажать клавишу ", чтобы сказать SMath Studio, что Вы собираетесь ввести текст. После этого, SMath Studio изменяет курсор в виде крестика в вертикальную линию. Контур окружает точку вставки, указывая, что Вы находитесь теперь в текстовой области. Этот контур называют текстовым окном. Для того чтобы ввести вторую линию текста, нажмите Shift + Ввод и продолжайте печатать текст.

Ранжированные переменные

Ранжированные переменные или переменные диапазона в Smath Studio являются разновидностью векторов и предназначены, главным образом, для создания циклов или итерационных вычислений. Простейший пример ранжированной переменной — это массив с числами, лежащими в некотором диапазоне с некоторым шагом.

SMath Studio позволяет сформировать вектор с заданным диапазоном значений. Для формирования такого вектора сначала нужно создать переменную диапазона. Например, сформируем вектор A для диапазона значений от 1 до 10 с шагом 0.5. Для этого нужно выполнить следующую последовательность действий [3]:

1. Наберите на клавиатуре A:=;
2. Наберите на клавиатуре 1, затем .. (диапазон значений с ПИ "Матрицы"), затем 10. Если вам необходимо, чтобы величина изменялась с шагом 1, можно на этом закончить. Если нужен шаг, отличный от единицы, то наберите на клавиатуре ; (точка с запятой) после 1, затем 1.5, диапазон значений и 10.

Определение пользовательских функций и использование встроенных функций

Система компьютерной математики Smath Studio имеет ряд встроенных функций, которые можно использовать при решении различных задач. Встроенные функции можно вводить вручную, либо использовать панель инструментов Функции. Но на этой панели приведены далеко не все имеющиеся в пакете функции. Поэтому есть ещё один способ доступа к функциям: Основное меню - Вставка - Функция [1].

Для справки по встроенным функциям (и их вводу) используется значок $f(x)$. Ряд встроенных функций после ввода открывающей скобки изменяют свой внешний вид и выглядят так же, как при письме «по бумаге». Перечислим некоторые встроенные алгебраические функции:

- abs(a) – модуль числа a;
- sqrt(a) – квадратный корень числа a;
- exp(a) – возвращает число e в степени a;
- ln(a) – натуральный логарифм числа a;
- lg(a) – десятичный логарифм числа a;
- xy2pol(x;y) – переводит координаты точки из декартовой системы в полярную;
- pol2xy(r;fi) – переводит координаты точки из полярной системы в декартову;
- random(n) – возвращает случайно выбранное натуральное число в диапазоне от 0 до n;
- m! – возвращает факториал числа m;

Кроме использования стандартных функций, пользователь может создавать свои собственные функции. Синтаксис описания пользовательской функции следующий:

- имя_функции (параметр1 , параметр2, .. , параметрN) :=выражение,
- где:
- имя_функции - название функции;
- параметр1 , параметр2, .. ,параметрN - параметры, с которыми будет вызываться функция;
- выражение - любое правильно написанное выражение, которое может использовать параметры функции.

Функции пользователя очень гибкий и удобный инструмент, они позволяют существенно уменьшить объем расчета путем замены часто повторяющихся участков. Но при работе с пользовательскими функциями стоит помнить, что

1. числу параметров, указанному при описании функции должно соответствовать число параметров при вызове.
2. параметрами функции могут быть Матрицы, Числа, и Строки. Но при вызове функции должен быть учтен порядок записи переменных. Т.е. если в описании функции 1я переменная - матрица, 2я - число, а 3я - строка, то и при вызове функции 1й параметр должен быть матрицей, 2й - числом, 3й - строкой.
3. в правой части описания функции могут использоваться элементы панели Программирование.

Арифметические операции. Целые и рациональные числа. Константы. Определение переменных в Scilab

Пользовательские переменные в Scilab

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 переменная перед использованием в выражении должна быть определена. В противном случае Scilab выдаст предупреждение об ошибке [2].

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 [2].

Если в течение одного сеанса присвоить некоторое значение ранее определенной в этом сеансе переменной, то в дальнейшем именно это значение будет использоваться программой во всех вычислениях, содержащих переменную. Говорят, что переменная переопределена [2].

Системные переменные и константы

При выполнении операции, Scilab обязательно присваивает ее результат какой-либо переменной, если в командной строке нет оператора присваивания, то результат будет присвоен системной переменной с именем ans. Переменную ans можно использовать в последующих вычислениях, но ее значение будет изменяться каждый раз после выполнения команды без оператора присваивания [2].

Ans — первый пример системной переменной. Имена других системных переменных в Scilab начинаются с символа %:

- %pi – число π (3.141592653589793);
- %e – число $e=2.7182818$.

Эти переменные используются в математических операциях в качестве констант. Их значения не могут быть изменены пользователем [2].

Математические операции в Scilab

Для выполнения простейших арифметических операций Scilab использует следующие операторы [2]:

- + 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.

При вводе выражения применена операция деления слева «\». Поскольку значение выражения не присвоено пользовательской переменной, то результат присваивается системной переменной ans. Если вычисляемое выражение длинное и желательно перенести его запись на следующую строку, то в конце незавершенной строки необходимо ввести три (или более) точки. После этого можно нажать «Enter» и продолжать набор оставшейся части на следующей строке [2].

Если после ввода команды нажать клавишу «Enter», то в рабочей области появится результат выполнения этой команды. Если результат отображать не нужно, то набор команды следует завершить символом «;» [2].

Правила работы с вещественными числами в Scilab

Числовые результаты могут быть представлены с плавающей (например, 3.2E - 6, 6.42E+2), или с фиксированной (например, 4.12, 6.05, 17.5489) точкой. Числа в формате с плавающей точкой представлены в экспоненциальной форме $mE\pm p$, где m - мантисса (целое или дробное число с десятичной точкой), p - порядок (целое число). Для того, чтобы перевести число в экспоненциальной форме к обычному представлению с фиксированной точкой, необходимо мантиссу умножить на десять в степени порядок [3].

При вводе вещественных чисел для отделения дробной части используется точка. Scilab в качестве результата выводит только восемь значащих цифр. Это формат вывода вещественного числа по умолчанию. Для того, чтобы контролировать количество выводимых на печать разрядов применяют команду `printf` с заданным форматом, который соответствует правилам принятым для этой команды в языке C.

Задание. Произвести вычисления с помощью пакетов Smath Studio и Scilab.

1. Вычислить значение выражения: $b = \frac{g^2 a - 3}{4g}$, $a = -1.57$, $g = -1.2$

2. Вычислить значение арифметического выражения. Установить формат результата: 5 знаков после запятой.

$$x^{-2} + \frac{y^2}{\sin^2(x) - \cos^2\left(\frac{x+1.15}{y-\sqrt{1.15}}\right)}, \quad x=1, \quad y=\frac{1}{2}$$

3. Рассчитать выражения в соответствии с вариантом. Ответ должен содержать m знаков после запятой; переменную x определить в соответствии с областью определения.

$$y = \frac{1 + \sin^2(8 + x^3)}{\sqrt[3]{8 + x^3}}, \quad m=4, \quad a=-5, \quad b=5, \quad h=1.$$

Лабораторная работа №3

Преобразование математических выражений

Цель работы: получить навыки работы с основными командами математических пакетов для символьного преобразования алгебраических выражений.

Символьные преобразования

Программа SMath Studio умеет выполнять некоторые символьные преобразования, такие как факторизация (выделение общего множителя), работа с показательными и логарифмическими выражениями, матрицами, взятие производных.

Для символьных преобразований необходимо записать выражение и нажать « \leftrightarrow » (либо Ctrl+.). Если ввести этот знак после математического выражения, ответ будет получен в виде обыкновенной дроби.

Задание. Осуществить преобразования математических выражений в пакете Smath Studio.

1. Упростить выражение: $(3\sin(x)+2\cos(x))^2+(2\sin(x)-3\cos(x))^2$.
2. Разложить число на простые множители: 1245.
3. Привести выражение к полиному по переменной y : $9+(3-z)y-2y(x+4y)$

Лабораторная работа №4

Решение уравнений и неравенств

Цель работы: получение навыков решения уравнений и неравенств с использованием математических пакетов Smath Studio и Scilab.

Решение уравнений и систем уравнений в пакете Smath Studio

Решение нелинейных уравнений

Для решения алгебраических уравнений с одним неизвестным в Smath Studio предусмотрена функция *solve*, которая, в зависимости от типа задачи, может включать либо два, либо четыре аргумента и, соответственно, использует разные алгоритмы поиска корней [1]:

1. *solve(уравнение, переменная)* - находит действительные корни уравнения относительно указанной переменной;
2. *solve(уравнение, переменная, число, число)* - находит действительные корни уравнения на промежутке [число, число] относительно указанной переменной.

Когда функция *solve* имеет четыре аргумента, следует помнить о двух её особенностях [1]:

1. внутри интервала может находиться более одного корня, тогда они будут найдены все;
2. значения $f(a)$ и $f(b)$ должны иметь разный знак.

Решение полиномиальных уравнений

Если функция $f(x)$ является полиномом, то все его корни можно определить, используя встроенную функцию *polyroots(v)*, где v - вектор, составленный из коэффициентов полинома [1].

Так как полином n -й степени имеет ровно n корней, вектор v должен состоять из $(n+1)$ элемента. В основе встроенной функции *polyroots(v)* лежат специальные численные алгоритмы, а результатом её действия является вектор, составленный из n корней рассматриваемого полинома [1].

Решение систем нелинейных уравнений

Решением системы нелинейных уравнений называется группа чисел, которые, будучи подставлены на место неизвестных, обращают каждое уравнение системы в тождество.

В Smath Studio для решения используют функцию *roots(уравнения; переменные)* [1].

Как уравнения, так и переменные задаются с помощью вектора (ПИ «Матрицы»). Уравнения можно задать непосредственно в первоначальном виде, где используют булево равно. Можно также предварительно привести уравнения к виду $f(x) = 0$. Функция *roots* находит только действительные решения.

Решение систем линейных уравнений

Один из методов решения систем линейных уравнений - метод обратной матрицы. Для решения систем таким способом используются операторы ПИ «Матрицы».

Также возможно решение систем линейных уравнений методом Крамера с использованием ПИ «Матрицы» и «Арифметика».

Решение уравнений и систем уравнений в Scilab

Решение алгебраических уравнений

Любое уравнение $P(x) = 0$, где $P(x)$ - это многочлен, отличный от нулевого, называется алгебраическим уравнением или полиномом. Всякое алгебраическое уравнение относительно x можно записать в виде $a_0x^n + a_1x^{n-1} + \dots + a_{n-1}x + a_n = 0$ [3].

Решение алгебраического уравнения в Scilab состоит из двух этапов. Необходимо задать полином $P(x)$ с помощью функции `poly`, а затем найти его корни, применив функцию `roots` [2,3].

Итак, определение полиномов в Scilab осуществляет функция `poly(a, "x ["fl"])`, где a - это число или матрица чисел, x - символьная переменная, fl - необязательная символьная переменная, определяющая способ задания полинома. Символьная переменная fl может принимать только два значения "roots" или "coeff" (соответственно "r" или "c"). Если $fl=c$, то будет сформирован полином с коэффициентами, хранящимися в параметре a . Если же $fl=r$, то значения параметра a воспринимаются функцией как корни, для которых необходимо рассчитать коэффициенты соответствующего полинома [2,3].

Функция `roots(p)` предназначена для решения алгебраического уравнения. Здесь p - это полином, созданный функцией `poly` и представляющий собой левую часть уравнения $P(x) = 0$ [2,3].

Трансцендентные уравнения

Уравнение $f(x) = 0$, в котором неизвестное входит в аргумент трансцендентных функций, называется трансцендентным уравнением. К трансцендентным уравнениям принадлежат показательные, логарифмические и тригонометрические. В общем случае аналитическое решение уравнения $f(x) = 0$ можно найти только для узкого класса функций. Чаще всего приходится решать это уравнение численными методами [3].

Численное решение нелинейного уравнения проводят в два этапа. В начале отделяют корни уравнения, т.е. находят достаточно тесные промежутки, в которых содержится только один корень. Эти промежутки называют интервалами изоляции корня, определить их можно, изобразив график функции $f(x)$ или любым другим методом. На втором этапе проводят уточнение отделенных корней, или, иначе говоря, находят корни с заданной точностью [2,3].

Для решения трансцендентных уравнений в Scilab применяют функцию `fsolve(x0,f)`, где $x0$ - начальное приближение, f - функция, описывающая левую часть уравнения $y(x) = 0$.

Системы уравнений

Если заданы m уравнений с n неизвестными и требуется найти последовательность из n чисел, которые одновременно удовлетворяют каждому из m уравнений, то говорят о системе уравнений. Для решения систем уравнений в Scilab также применяют функцию `fsolve(x0,f)` [3].

Задание. Решить уравнения в пакетах Smath Studio и Scilab.

$$\frac{x+2}{x+1} + \frac{2-x}{1-x} + \frac{4}{x-1}$$

1. Дано уравнение $\frac{x+2}{x+1} + \frac{2-x}{1-x} + \frac{4}{x-1}$. Определить один из корней с помощью задания начального значения, другой с помощью задания интервала, которому принадлежит корень. Найти вектор корней уравнения. Все вычисления производить с точностью 0.00001.

2. Найти все корни полиномиального уравнения: $2x^4 - 21x^3 + 74x^2 - 105x + 50 = 0$.

3. Найти одно из решений системы уравнений численно:
$$\begin{cases} \cos(x) + y = 1.5 \\ 2x - \sin(y - 0.5) = 1 \end{cases}$$

4. Решить систему линейных уравнений:

$$\begin{cases} 0.65x_1 - 0.93x_2 + 0.45x_3 = -0.72 \\ 1.15x_1 + 0.43x_2 - 0.72x_3 = 1.24 \\ 0.56x_1 - 0.18x_2 + 1.03x_3 = 2.15 \end{cases} .$$

Лабораторная работа №5

Построение 2D и 3D графиков

Цель работы: получение навыков работы по созданию двух- и трехмерных графиков в математических пакетах.

Построение 2D и 3D графиков в Smath Studio

Построение двумерных графиков

Графики функции в Smath Studio бывают 2х видов: двумерные (2d) и трехмерные (3d). Для работы с ними есть специальная панель "График". Вставить график в расчет можно несколькими способами:

- 2d график можно вставить комбинацией shift + @;
- при помощи меню Вставка - График.

Таким образом, при создании графика необходимо выбрать в меню Вставка – График – Двумерный (2D) или нажать shift + @. В нижней части появившегося окна вводится имя функции или сама функция. Аргументом должна быть только переменная «x» (т.е. функция имеет вид $f(x)$).

Чтобы построить графики нескольких функций в одной системе координат, необходимо добавить фигурные скобки, нажать пробел, растянуть их за чёрную точку в правом нижнем углу до требуемого размера и затем ввести функции.

Для изменения окна графика необходимо растянуть его за одну из трёх чёрных точек. Вращение ролика мыши приводит к масштабированию всего графика в целом. Для масштабирования по оси абсцисс необходимо при вращении ролика держать нажатой клавишу Shift, по оси ординат – Ctrl. Движение мыши при нажатой левой клавише ведёт к разному результату в зависимости от выбранного значка на панели: перемещает систему координат («Перемещать», «Вращать», включено по умолчанию), либо масштабирует график («Масштабировать»). Пункты «График точками» и «График линиями» изменяют соответствующий вид отображения. Нажатие «Обновить» сбрасывает все сделанные изменения.

Стоит обратить внимание, что в объявлении функции можно использовать любые имена переменных. Но график строится только относительно "x" и "y". Т.е. внутри графика функция должна быть вызвана с использованием этих переменных.

Построение трёхмерных графиков

Трёхмерный график строится через меню Вставка – График – Трёхмерный. В нижней части окна нужно ввести имя функции или саму функцию. Аргументами должны быть только «x» и «y». Графики нескольких функций также строятся с помощью фигурных скобок.

Для форматирования графика нужно сделать его активным и установить желаемый размер окна. Масштабирование вращением ролика осуществляется для всех осей одновременно. В зависимости от выбранного значка на панели график можно движением мыши вращать (по умолчанию), масштабировать или перемещать. Также доступен выбор отображения линиями или точками. Нажатие «Обновить» сбрасывает все сделанные изменения.

Построение графика по точкам

Для построения графика по точкам нужно использовать матрицы. При построении двумерных графиков, матрица содержит пары координат X и Y. Для трехмерных - эта

матрица содержит тройки координат X, Y, Z. Матрица такого формата способна отображаться лишь на 3D-графике.

Такой подход к построению графиков как правило требует программирования. Для работы с матрицами могут понадобиться функции `augment`, `el(2)`, `el(3)`, `stack`.

Построение 2D и 3D графиков в Scilab Построение двумерных графиков Функция `plot`

Для построения графиков вида $y=f(x)$ в Scilab существует функция `plot`. Эта функция была значительно модифицирована начиная с 4-й версии пакета. Так, при простейшем обращении к функции `plot(x,y)` будет создано окно, в котором будет построен график функции $y(x)$ [3].

Если повторно обратиться к функции `plot`, то будет создано новое графическое окно и в нем будет построен новый график. Для построения нескольких графиков в одной системе координат можно поступить одним из следующих способов:
1. Обратиться к функции `plot` следующим образом `plot(x1,y1,x2,y2,...xn,yn)`, где x_1, y_1 – массивы абсцисс и ординат первого графика; x_2, y_2 – массивы абсцисс и ординат второго графика; ..., x_n, y_n – массивы абсцисс и ординат n-ого графика.

Каждый график изображать с помощью функции `plot(x,y)`, но перед обращением к функциям `plot(x2,y2)`, `plot(x3,y3)`, ..., `plot(xn,yn)` вызвать команду `mtlb_hold(' on')`, которая блокирует режим создания нового окна [2,3].

Обратите внимание, что при построении графиков первым способом Scilab автоматически изменяет цвета изображаемых в одной системе координат графиков. Однако управлять цветом и видом каждого из изображаемых графиков может и пользователь, для чего необходимо воспользоваться полной формой функции `plot`: `plot(x1, y1, s1, x2, y2, s2, ..., xn, yn, sn)`, где x_1, x_2, \dots, x_n – массивы абсцисс графиков; y_1, y_2, \dots, y_n – массивы ординат графиков; s_1, s_2, \dots, s_n – строка, состоящая из трех символов, которые определяют цвет линии, тип маркера и тип линии графиков (см. табл. 4.1-4.3), в строке могут использоваться один или два символа [2,3].

Параметры функции `plot`. Символы цветов линий

- `y` - желтый;
- `m` - розовый;
- `c` - голубой;
- `r` - красный;
- `g` - зеленый;
- `b` - синий;
- `w` - белый;
- `k` - черный;

Параметры функции `plot`. Символы типов маркера:

- `.` - точка;
- `o` - кружок;
- `x` - крестик;
- `+` - знак "плюс";
- `*` - звездочка;
- `s` - квадрат;
- `d` - ромб;
- `v` - треугольник вершиной вниз;
- `^` - треугольник вершиной вверх;
- `<` - треугольник вершиной влево;
- `>` - треугольник вершиной вправо;
- `p` - пятиконечная звезда;
- `h` - шестиконечная звезда;

Параметры функции *plot*. Символы типов линий.

- - сплошная;
- : пунктирная;
- -. штрихпунктирная;
- -- штриховая.

Функция *plot2d*

В общем виде обращение к функции имеет вид [3]: *plot2d*([*loglog*],*x*,*y*,[*key1=value1,key2=value2, ..., keyn=valuen*], где

- *logflag* – строка из двух символов, каждый из которых определяет тип осей (*n*- нормальная ось, *l* – логарифмическая ось), по умолчанию "nn";

- *x* – массив абсцисс;
- *y* – массив ординат(или матрица, каждый столбец которого содержит массив ординат очередного графика) (количество элементов в массиве *x* и *y* должно быть одинаковым), если *x* и *y* – являются матрицами одного размера, то в этом случае, каждый столбец матрицы *y* отображается относительно соответствующего столбца матрицы *x*;

- *keyi=valuei* – последовательность значений параметров графиков, возможны следующие значения параметров: *style* – определяет массив (*mas*) числовых значений цветов графика (*id* цвета), количество элементов массива совпадает с количеством изображаемых графиков, по умолчанию, по умолчанию представляет собой массив *masi=i*, цвет *i*-й линии совпадает с номером *i*, для формирования *id* соответствующего цвета (кода цвета) можно воспользоваться функцией *color*, которая по названию (*color*("цвет")) или коду *grb* (*color*(*r,g,b*)) цвета формирует нужный *id* (код) цвета. Если значение стиля отрицательное то это будет точечный график без соединения точек между собой линиями.

- *rect* – этот вектор [*xmin, ymin, xmax, ymax*] определяет размер окна вокруг графика;

- *frameflag* – параметр определяет окно в котором, будет изображаться график, он может принимать следующие значения: 0 – не вычислять размеры окна, использовать значения по умолчанию или значения из предыдущего графика, 1 – размер окна определяется параметром *rect*, 2 – размер окна определяется из соотношения между минимальным или максимальным значениями *x* и *y*, 3 – размер окна определяется параметром *rect* в в изометрическом масштабе, 4– размер окна определяется из соотношения между минимальным или максимальным значениями *x* и *y* в изометрическом масштабе;

- *axesflag* - параметр, который определяет рамку вокруг графика, следует выделить следующие значения этого параметра: 0 – нет рамки вокруг графика; 1 – изображение рамки, ось *y* слева; 3 – изображение рамки, ось *y* справа; 5 – изображение осей проходящих через точку (0,0);

- *paх* – этот параметр используют, если параметр *axesflag* равен 1, *paх* представляет массив из четырех значений [*nx, Nx, ny, Ny*] – где *Nx* (*Ny*) – число основных делений с подписями под осью *X* (*Y*), *nx* (*ny*) – число промежуточных делений;

- *leg* – строка, определяющая легенды для каждого графика, структура строки такая: "*leg1@leg2@leg3@...@legn*", где *leg1* – легенда первого графика, ..., *legn* – легенда первого графика.

Пример построения нескольких графиков различного цвета приведен ниже:

```
x=[-2*%pi:0.1:2*%pi];
y=[sin(x); cos(x)];
plot2d(x,y',style=[color("red"), color("blue")]);
```

Функция *plot2d*

Функцию *plot2d* [3] можно использовать для построения точечных графиков. В этом случае обращение к функции имеет вид *plot2d*(*x,y,d*), здесь *d* – отрицательное число, определяющее тип маркера.

Оформление графиков

Рассмотрим основные возможности Scilab по оформлению графиков.

Изображение сетки на графике. Для изображения сетки следует воспользоваться функцией `xgrid(color)`, `color` определяет id цвета линии сетки.

Заголовки на графике

Для вывода заголовков на графике служит функция `xtitle(title, xstr, ystr)`, здесь `title` – название графика, `xstr` – название оси X, `ystr` – название оси Y.

Нанесение легенд на график

Для нанесения легенд на график служит функция `legend(leg1,leg2,...,legn[,pos][,boxed])`, где `leg1` – имя первой легенды, `leg2` – имя второй легенды, ..., `legn` – имя n-й легенды; `pos` – месторасположение легенды: 1 – верхнем правом углу (по умолчанию), 2 – верхнем левом углу, 3 – нижнем левом углу, 4 – нижнем правом углу, 5 – определяется пользователем после изображения графика; `boxed` – логическая переменная (по умолчанию %t), которая определяет рисовать или нет рамку вокруг легенды.

Кроме того, Scilab позволяет после вывода графика перейти в режим форматирования с помощью команды `Edit – Figure proterties` в окне график.

Построение трехмерных графиков в Scilab

В целом процесс построения трехмерного графика можно разделить на три этапа [3]:

1. Создание в области построения графика прямоугольной сетки.
2. Вычисление значений функции во всех узлах сетки.
3. Обращение к функции построения трехмерной графики.

Функция plot3d и plot3d1

В Scilab поверхность можно построить с помощью функций `plot3d` или `plot3d1`. Их отличие состоит в том, что `plot3d` строит поверхность и заливает ее одним цветом, а `plot3d1` строит поверхность, каждая ячейка которой имеет цвет, зависящий от значения функции в каждом соответствующем узле сетки.

Обращение к функциям следующее:

`plot3d(x,y,z,[theta,alpha,leg,flag,ebox][keyn=valuen]),`

`plot3d1(x,y,z,[theta,alpha,leg,flag,ebox][keyn=valuen]),`

здесь `x` – вектор -столбец значений абсцисс;

`y` – вектор -столбец значений ординат;

`z` – матрица значений функции;

`theta, alpha` – действительные числа, которые определяют в градусах сферические координаты угла зрения на график. Попросту говоря, это угол, под которым наблюдатель видит отображаемую поверхность;

`leg` – подписи координатных осей графика;

`flag` – массив, состоящий из 3 целочисленных параметров [`mode,type,box`]. Здесь `mode` – устанавливает цвет поверхности.

`ebox` – определяет границы области, в которую будет выводиться поверхность, как вектор [`xmin,xmax,ymin,ymax,zmin,zmax`]. Этот параметр может использоваться только при значении параметра `type=1`;

`keyn=valuen` – последовательность значений свойств графика `key1=value1, key2=value2,...,keyn=valuen`, таких как толщина линии, ее цвет, цвет заливки фона графического окна, наличие маркера и др.

Таким образом, функции `plot3d` (`plot3d1`) в качестве параметров необходимо передать прямоугольную сетку и матрицу значений в узлах сетки.

Задание. Построить графики приведенных ниже функций.

1. Постройте графики функций.

Функция одной переменной	Полярная система координат	Функция двух переменных
$y = \frac{3}{x^3} + \frac{2}{x^2} + \frac{1}{x}$	$x = t^3 - 3\pi$ $y = t^3 - 6 \cdot \arctg(t)$	$z = \sin\left(\frac{x}{y}\right) \cos\left(\frac{y}{x}\right)$

2. Отобразить графически пересечение поверхностей

$$f_1(x, y) := \frac{(x+y)^2}{10} \quad \text{и}$$

$$f_2(x, y) := 5 \cos\left(\frac{x-y}{3}\right).$$

Лабораторная работа №6

Дифференциальное и интегральное исчисление

Цель работы: получение навыков нахождения производных функций и решения интегралов с использованием математических пакетов.

Дифференциальное и интегральное исчисление в Scilab

Интегрирование в Scilab

Вычислить определенный интеграл в Scilab можно при помощи функции `int(a, b, f)`, где a и b — нижний и верхний пределы интегрирования соответственно, f — имя подынтегральной функции [2].

Нахождение интеграла будет выглядеть следующим образом [2]:

```
-->function y=f(x),y=3*cos(x/2) endfunction
-->intg(0,%pi,f)
ans =6.
```

Вычисление производной в Scilab

В Scilab можно вычислять производную функции в заданной точке. Вычисление происходит при помощи команды `numdiff(f,x0)`, где f — имя дифференцируемой функции переменной x , x_0 — координата точки в которой необходимо вычислить производную [2].

Задание. Найти производные функций и вычислить значения заданных интегралов.

1. Дана функция $f(x) = \frac{x}{2x-1}$, найти $f'(0)$, $f''(1)$, $f'''(2)$.

2. Вычислить определенные интегралы

$$\int_0^{2\pi} \sin x \, dx; \quad \int_0^{+\infty} e^{-x} \, dx; \quad \int_0^1 \ln^2 x \, dx; \quad \int_0^a x^2 \sqrt{a^2 - x^2} \, dx; \quad \int_{-1}^1 x^2 e^{-x} \, dx.$$

Раздел 2. Самостоятельная работа

Вид самостоятельной работы:

1. Проработка лекционного материала по следующим темам:

- Математические пакеты Smath Studio и Scilab
- Структура окон Smath Studio и Scilab
- Арифметические операции. Целые и рациональные числа, константы. Синтаксис команд. Стандартные функции

- Преобразование математических выражений
 - Решение уравнений и неравенств
 - Построение 2D и 3D графиков
 - Дифференциальное и интегральное исчисление
 - Решение обыкновенных дифференциальных уравнений в пакетах Smath Studio и Scilab
2. Оформление отчетов по лабораторным работам:
- Структура окон Smath Studio и Scilab.
 - Арифметические операции. Целые и рациональные числа, константы. Синтаксис команд. Стандартные функции.
 - Преобразование математических выражений.
 - Построение 2D и 3D графиков.
 - Дифференциальное и интегральное исчисление.
3. Самостоятельное изучение тем:
- Решение обыкновенных дифференциальных уравнений в пакетах Smath Studio и Scilab
 - Решение дифференциальных уравнений в частных производных в пакетах Smath Studio и Scilab
 - Программирование в пакетах Smath Studio и Scilab

Список литературы

1. Квасов, Б.И. Численные методы анализа и линейной алгебры. Использование Matlab и Scilab [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : Лань, 2016. — 328 с., дата обращения: 11.05.2018: В другом месте, <https://e.lanbook.com/book/71713>.
2. Ощепков, А.Ю. Системы автоматического управления: теория, применение, моделирование в MATLAB [Электронный ресурс] : учеб. пособие — Электрон. дан. — Санкт-Петербург : Лань, 2013. — 208 с., дата обращения: 11.05.2018: В другом месте, <https://e.lanbook.com/book/5848>