Бибина Виктория Николаевна

МГПИ им. М.Е. Евсевьева

г. Саранск

Студент

Физико-математический факультет

Кафедра информатики и вычислительной техники

**3D ГРАФИКИ В SCILAB**

Выполнил: студент группы

МДМ-114 Бибина В.Н.

Проверила: Кормилицына Т.В.

**Содержание**

Введение

1 Возможности Sсilab

2 Основы программирования в Scilab

3 Построение графиков функций

Заключение

Список использованных источников

## Введение

Scilab – пакет прикладных математических программ, предоставляющий открытое окружение для инженерных (технических) и научных расчётов.

Решение математических задач с помощью компьютерных технологий до сих пор является актуальной темой в области информатики и математики. Программа Scilab предназначена для выполнения инженерных и научных вычислений.

Scilab представляет собой язык программирования высокого уровня, который широко используется во всем мире для технического и научного применения. Scilab был разработан в Национальном исследовательском институте информатики и автоматизации (Institut national de recherche en informatique et enautomatique, INRIA) и Национальной школе дорожного ведомства (École Nationale des Ponts et Chaussées, ENPC),в 1994 году во Франции. И с 2003 года поддержкой Scilab занимается консорциум Scilab Consortium [13].

В состав пакета Scilab входит большая библиотека математических функций. Для пакета Scilab существует масса наборов инструментальных средств, реализованных сообществом пользователей. Они позволяют строить двумерные и трехмерные графики и анимацию, проводить оптимизацию, статистические вычисления, строить графы и сети, выполнять обработку сигналов, моделировать динамические системы, а также открывают множество других возможностей [21].

Изучение возможностей системы компьютерной математики Scilab для решения математических задач и определяет ***актуальность данного исследования.***

**1 Возможности Scilab**

Scilab – пакет прикладных математических программ, предоставляющий открытое окружение для инженерных (технических) и научных расчётов [3].

Распространение пакета Scilab началось с 1994 года вместе с исходным кодом через интернет. В 2003 году для поддержки Scilab был создан консорциум Scilab Consortium. Сейчас в него входят 25 участников, в том числе Mandriva, INRIA и ENPC (Франция).

Отличительными особенностями, которого являются: бесплатность, свободность использования (с версии 5.0), маленький размер (20 МБ, с 5.4.1 версии Silab увеличился в объеме до 117 МБ), также возможность запуска в консоли без использования графического интерфейса, это позволяет производить автоматизированные вычисления, есть пакетный режим.

Scilab можно использовать на большинстве систем UNIX, а также на современных операционных системах Windows. Так же как GNU Octave, пакет Scilab обладает отличной документацией. Поскольку это европейский проект, вы можете найти документацию и статьи не только на английском, но и на множестве других языков. В состав пакета Scilab входит множество математических функций, к тому же он очень хорош для обработки сигналов [1].

Scilab был спроектирован так, чтобы быть открытой системой, где пользователи могут добавлять свои типы данных и операции над этими данными путем перегрузки.

Пакет разработан Scilab Group INRIA-Rocquencourt Metalau Project. Свободно распространяемую версию пакета вместе с полной документацией на английском языке в формате pdf можно получить по адресу http://www.scilab.org/. Современные версии готовит Consrtium Scilab (INRIA, ENPC). Web страница в Интернете – www.scilab.org.

Scilab = Science Laboratory (Научная Лаборатория).

Основные характеристики пакета Scilab

1.Распространение: Scilab свободно распространяется вместе с исходными кодами. Использование, копирование, изменение, распространение - свободные. Пакет защищен специальной лицензией, основное отличие которой от стандартной GNU лицензии, по утверждению авторов, определяется стремлением избежать появления клонов.

2.Платформы: UNIX (включая Linux), Windows. Исходные тексты, рабочая версия для Windows и документация доступны в сети.

3.Имеется интерфейс и система помощи.

4.Имеются алгоритмы базовой математики.

5.Имеется возможность программирования.

6.Имеется возможность работать не только в численном виде, но и в формульном варианте.

7.Имеется возможность работы с графикой.

8.Интерфейс с прикладными программами: имеется возможность использовать откомпилированные функции языков Си и Фортран.

В Scilab еализованы численные методы решения задач вычислительной математики. Пакет прикладных программ Scilab содержит много математических функций. Также имеются разнообразные структуры данных: списки, полиномы, рациональные функции, линейные системы, интерпретатор и язык высокого уровня. Пакет поддерживает основные элементарные и множество специальных функций, применяемых в математике [29].

Так выделим возможности Scilab:

1. Решение задач линейной алгебры. Массивы и матрицы.
2. Построение двумерных графиков.
3. Построение трехмерных графиков.
4. Решение нелинейных уравнений и систем уравнений.
5. Решение дифференциальных уравнений.
6. Численное интегрирование и дифференцирование.
7. Программирование в Scilab.
8. Создание графических приложений в среде Scilab.
9. Обработка экспериментальных данных.
10. Решение задач оптимизации.

Также пакет Scilab поддерживает основные элементарные и множество специальных функций, применяемых в математике.

Итак, система Scilab обладает большими возможностями в решении математических задач.

**2 Основы программирования в Scilab**

Рассмотрим основные теоретические сведения пакета Scilab. При запуске пакета Scilab открывается основное окно (см. рис. 1), которое условно можно разделить на две области: это область меню, которая расположена вверху экрана и панель инструментов, а также рабочая область с командной строкой, в которой, собственно, и происходит решение задачи.

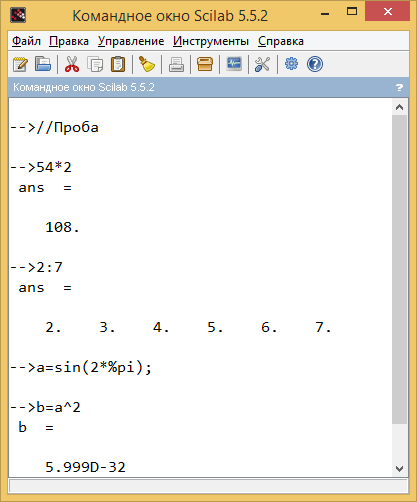


Рисунок 1 – Основное окно Scilab 5.5.2

Индикатором ввода в командной строке является символ**-->**в рабочей области, возле которого находится курсор. Ввод завершается нажатием клавиши *Enter*.

Пока не нажата клавиша *Enter*, текст в командной строке можно редактировать стандартным способом. После нажатия *Enter* команда перемещается в область просмотра доступ к ее редактированию закрыт. Если есть необходимость вернуться к этой или другой данной команде, то при помощи клавиши управления курсором ***↑*** и ***↓*** ранее выполненные команды могут быть возвращены в командную строку. После этого их можно редактировать и выполнять повторно [23].

Для выполнения простейших арифметических операций Scilab использует следующие операторы, представленные в таблице 1.

Таблица 1 – Арифметические операции Scilab.

|  |  |
| --- | --- |
| + | сложение; |
| - | вычитание; |
| \* | умножение; |
| / | деление слева на право; |
| \ | деление справа на лево; |
| ^ | возведение в степень; |

Чтобы вычислить значение арифметического выражения, необходимо ввести его в командную строку и нажать *Enter*.

Рассмотрим переменные. Пользователь может определить переменную, чтобы использовать ее в последующих расчетах. Определить переменную означает задать ее имя и значение. С этой целью используется *оператор присваивания*, в общем случае имеющий вид: *имя\_переменной = значение\_переменной.*

В Scilab допускается использование в имени переменной множества символов. Система различает большие и малые буквы в именах переменных, это имена разных переменных [8].

Выражение в правой части *оператора присваивания* может быть числом, арифметическим выражением, символьным выражением или строкой символов. В последнем случае выражение справа берется в *кавычки*.

Также существует системная переменная с именем *ans*(от англ. *answer — ответ*), если пользователь не задал в командной строке имя переменной.

Любой математический пакет или язык программирования оперирует двумя типами функций: *встроенными и определенными пользователем*.

В общем виде обращение к функции имеет вид: *имя \_ переменной = имя \_ функции (арг1 [,арг2,...]),* где *имя \_ переменной* – переменная, куда запишутся результаты работы функции; *имя \_ функции –* имя встроенной или введенной ранее пользователем функции; *арг1, арг2*  – аргументы функции [8].

Встроенные функций в Scilab вполне достаточно для проведения, анализа и оформления инженерных вычислений. Их рассмотрение начнем с математических функций, как наиболее распространенных в инженерной практики представленных в таблице А.1(см. Приложение А).

Рассмотрим пример 1.При x=1.34 и y=0.45 вычислить . Ниже представлен листинг программы вычисления примера 1 рисунок 2.

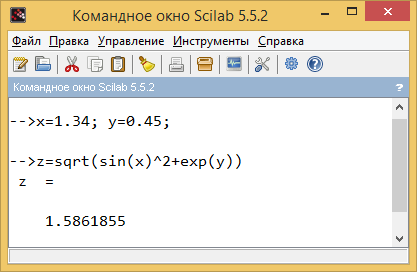


Рисунок 2 – Листинг программы вычисления примера 1

В Scilab функция f(x) можно оформляется как *функция пользователя* двумя способами.

Способ 1. Применение оператора *deff*, имеющего в общем случае вид:

*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\_М*–входные параметры,*имя1=выражение1;...;имя N= выражение N* – определение выходных переменных [8].

Способ 2. Применение конструкции *function*, синтаксис которой следующ: *function [имя1,...,имяN] = имя \_ функции (переменная\_1,..., переменная \_ М)...endfunction,* где *имя1,...,имя N* – список выходных параметров; *имя\_ функции* – имя с которым эта функция будет вызываться, *переменная\_1, ...,переменная\_ M* – входные параметры. Определение выходных переменных происходит в теле функции [8].

Рассмотрим пример 2. Определите функцию и вычислите ее значение в точках {-1,3; 2,1; 3,7}.Ниже представлен листинг программы вычисления примера 2 рисунок 3.

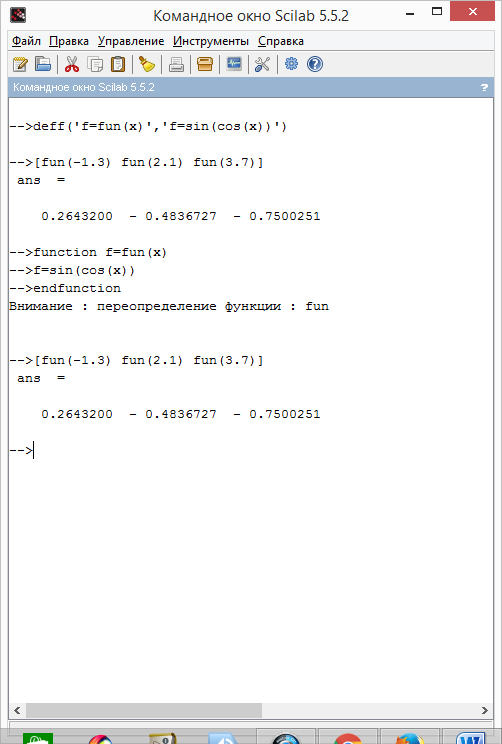


Рисунок 3 – Листинг программы вычисления примера 2

Пример 3. Определите функцию, в которой одновременно задаются и вычислите ее значение в точке (-2,5; 1,7).Далее представлен листинг программы вычисления примера 3 рисунок 4.

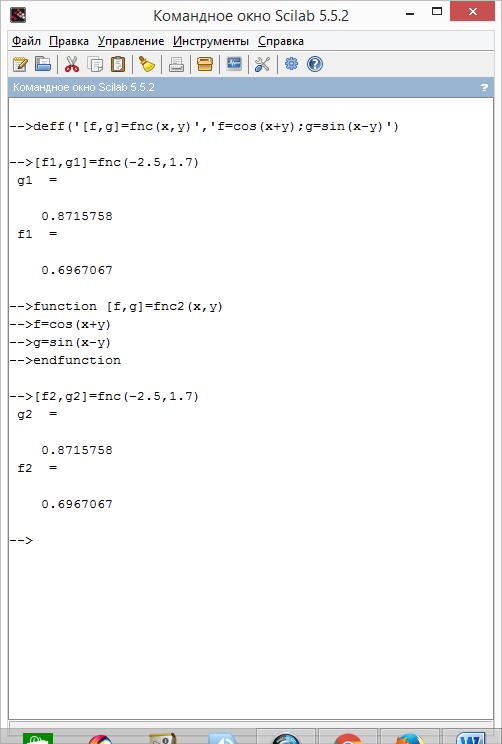


Рисунок4 – Листинг программы вычисления примера 3

При решении задач часто приходится иметь дело с алгоритмом ветвления – ход вычисления зависит от выполнения или невыполнения какого-либо условия: *если ... то... иначе (в противном случае).*

Одним из основных операторов, реализующих ветвление в большинстве языков программирования, является условный оператор *if*. Обычный оператор *if* работает по следующем алгоритму: если условие истинно, то выполняются *операторы1*, если ложно – *операторы2* [23].

Пример 4. Заданы два числа a и b. Вычислить их сумму, если a>b и произведение в противном случае (листинг программы вычисления примера 4 представлен на рисунке A.1 (см. Приложение А).

Для задания условий используются операторы, представленные в таблице 2.

Таблица 2 – Операторы для задания условий.

|  |  |
| --- | --- |
| *&(and)* | логическое «и» |
| *|(or)* | логическое «или» |
| **~** *(not)* | логическое отрицание |
| **<** | Меньше |
| **>** | Больше |
| **>=** | больше или равно |
| **<=** | меньше или равно |

В случае сложных условий составляющие его элементарные условия заключаются в скобки.

Пример 5. Заданы два числа a и b. Вычислить их сумму, если a>0 и b>0 одновременно и произведение в противном случае*.* Представлен листинг программы решение примера 5 на рисунке А.2 (см. Приложения А).

Что касается же графиков функций, Scilab умеет рисовать графики функций на плоскости (кривые) или в трехмерном пространстве (поверхности), основные геометрические шаблоны (прямоугольники, эллипсы), строить некоторые специальные диаграммы (гистограммы, графики, применяемые в теории автоматического контроля, и т. д.) и многое другое. Предусмотрена широкая настройка свойств, допускается определять цвета, заливки, точку обзора (в трехмерном случае), отображать сетки и управлять десятками других характеристик. Графики можно выводить в одном окне, сохранять во внешних файлах, наносить на них пояснительные надписи, задавая цвет, размер и гарнитуру шрифта [18].

*Обзор графических возможностей Scilab*

Scilab предоставляет возможности для создания различных типов графиков, среди которых:

* двухмерные графики: *plot*,
* контурные графики: *contour*,
* трёхмерные графики: *surf*
* гистограммы: *histplot*
* столбиковые диаграммы: *bar*

Для того чтобы увидеть пример трехмерного графика, достаточно набрать в консоли Scilab команду

При создании графиков в данном разделе используются вспомогательные функции, приведенные в таблице 3 .

Таблица 3 – Функции, используемые при построении графиков

|  |  |
| --- | --- |
| linspace | генерирует вектор из заданного числа равноотстоящих значений |
| feval | вычисляет значения функции в точках сетки |
| legend | задает легенду текущего графика |
| title | отображает название на текущем графике |
| xtitle | отображает название и подписи к осям на текущем графике |

Любой график имеет следующие элементы:

* Собственно сам график функции (один или несколько), включая параметры его отображения;
* Оси системы координат, включая внешний вид, масштаб отсчета и подписи осей;
* Координатная сетка;
* Подпись графика;
* Легенда;
* Надписи на самом графике.

При вызове непосредственно функций построения, многие второстепенные элементы строятся автоматически по заранее заготовленным шаблонам, однако, всеми этими элементами можно управлять.

Функция *plot* предназначена для построения двумерных графиков функции одной переменной вида.

*Для построения одномерных функций в прямоугольной системе координат служат следующие функции*:

семейство функций *plot2d(),* выполняющих построение по точечной функции, где символ звездочки кодирует заранее определенный шаблон:

* *plot2d* — строит график по форматам, определяемым пользователем;
* *plot2d2* — строит график в виде ступенчатой функции;
* *plot2d3* — строит график в виде вертикальных полосок;
* *plot2d4* — строит график с указанием направления;

Для начала попробуем построить параболу через точечную функцию. Листинг программы представлен на рисунок 5. Для этого определим квадратичную функцию: .

Теперь нам нужны точки. Для примера определим два вектора в один из которых мы запишем значения аргументов в диапазоне от -10 до 10, а во второй вектор — значения функции от этих аргументов: *;.*

Самый простой вызов функции *plotd()*. Если до этого ни одного графического окна не было создано, то оно будет создано автоматически. Вызовите функцию. Мы увидим самый простой график рисунок 6 [3].

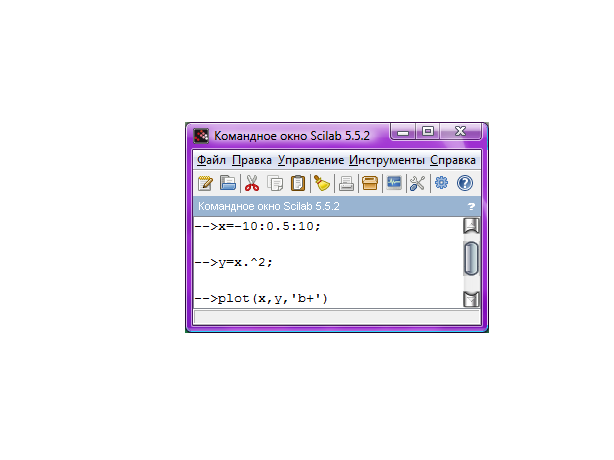


Рисунок 5 – Листинг программы построения графика функции параболы.

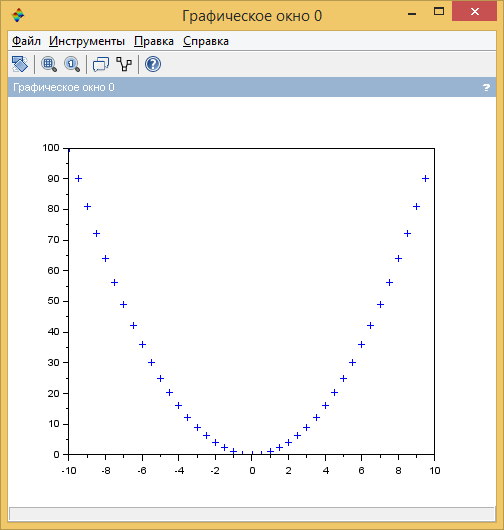


Рисунок 6 – График функции параболы

**Построение трехмерной линии, заданной параметрически**

В качестве примера рассмотрим построение трехмерной линий, заданной уравнением:

xset('window',20)

t=[0:0.1:5\*%pi]';

param3d1([sin(t),sin(2\*t)],[cos(t),cos(2\*t)],..

list([t/10,sin(t)],[3,2]),35,45,"X@Y@Z",[2,3])

xdel(20);

a=gca();

t=[0:0.1:5\*%pi]';

param3d1([sin(t),sin(2\*t)],[cos(t),cos(2\*t)],[t/10,sin(t)]);

a.rotation\_angles=[65,75];

a.data\_bounds=[-1,-1,-1;1,1,2];

a.thickness = 2;

h=a.children;

h.children(1).foreground = 3;

curve2 = h.children(2);

curve2.foreground = 6;

curve2.mark\_style = 2;

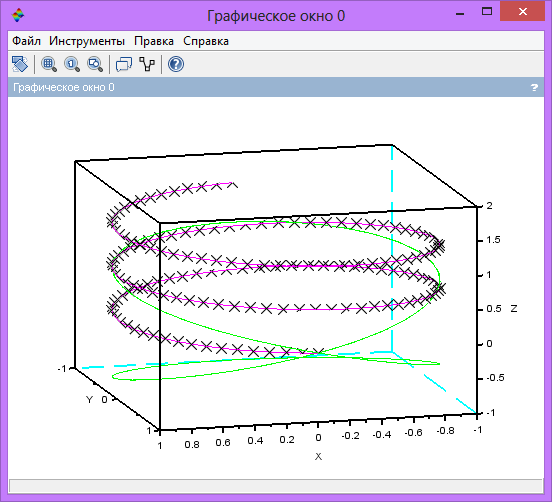


Рисунок 7 – График функции

Вид графика можно изменять, добавив при обращении к функцииplot помимо основных аргументов еще один аргумент — строку, состоящую из трех символов, которые будут определять цвет линии, тип маркера и тип линии графика. Обращение к функции plot будет выглядеть:*plot(x,y,string)*, в случае, если необходимо построить один график, или *plot (x1,y1,string1,x2,y2, string2,x3,y3, string3...xn,yn, stringn)* — для n графиков. Строка string выглядит следующим образом *'параметр 1 параметр 2 параметр 3'* [1]*.*

Символы пишутся один за другим без разделителей.

1. Параметр 1 определяет цвет линии графика. Значения параметра функции plot, определяющего цвет графика представленных в таблице А.2.(см. Приложение А).

2. Параметр 2 определяет тип маркера графика. Значения параметра, определяющего тип маркеров (точек) графика представленных в таблице А. 3.(см. Приложение А).

3. Параметр 3 определяет тип линии графика. Значения параметра, определяющего тип линии графика представленные в таблице А.4 (см. Приложение А).

Поверхности строятся следующими функциями:

* *plot3d()* — строит поверхность по координатам;
* *plot3d1()* — функция с расширенными возможностями настройки цвета;
* *plot3d3* — строит сетку четырехсторонними полигонами;
* *surf()* — строит поверхность;
* *mesh()*–– строятся поверхности.

График — это графическое представление функции. В среде Scilab все графики строятся по точкам, при этом каждые две соседние точки соединяются друг с другом отрезком.

Чем меньше расстояние между точками, тем меньше искажение графика за счет замены его реального образа кусочно-линейной функцией.

К сожалению, координаты точек нужно где-то хранить, что является главным ограничивающим фактором в желании получить как можно более плавное представление.

Внешний вид графика определяют следующие факторы:

Математическая функция и система координат:

**Двухмерный график** — график функции с одним аргументом, либо, если функция представлена неявно, одномерной функции;

* в прямоугольной системе координат;
* в полярной системе координат;
* в комплексной плоскости (например, годографы);
* в векторной плоскости;

**Трехмерный график** — график функции с двумя аргументами, либо, если функция представлена неявно, двухмерной функции;

* в пространственной прямоугольной системе координат;
* в сферической системе координат;
* в цилиндрической системе координат;
* в векторном пространстве.

Способ представления:

* точечный график;
* линейный график (с учетом всевозможных представлений линии);
* полигональный график;
* градиентный график;
* контурный график
* гистограмма.

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 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 ;

Здесь

**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 символы, отделяемые знаком @. Например , 'X@Y@Z'.

**flag** – массив, состоящий из 3 целочисленных параметров [mode,type,box].

Здесь **mode** устанавливает цвет поверхности. (По умолчанию, равен 2 - цвет заливки синий ,прямоугольная сетка в ыводится )

**Type** - позволяет управлять масштабом графика, по умолчанию имеет значение 2;

**box** - определяет наличие рамки вокруг отображаемого графика. По умолчанию равен 4.

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

Здесь **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.

**3 Построение графиков функций**

Приведем пример построения графика функции .

Для формирования сетки воспользуемся функцией meshgrid, запишем значения аргументов x, y. Затем обратимся к функции surf. Листинг программы представлен на рисунке 7. Далее сам график функции на рисунке 8.

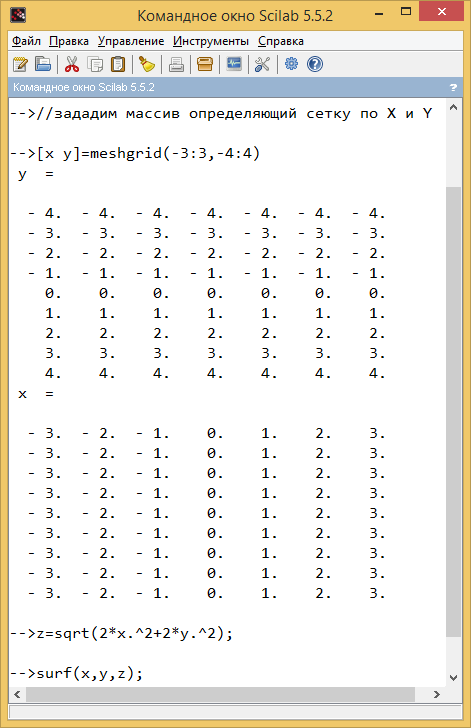


Рисунок 7 – График функции параболы

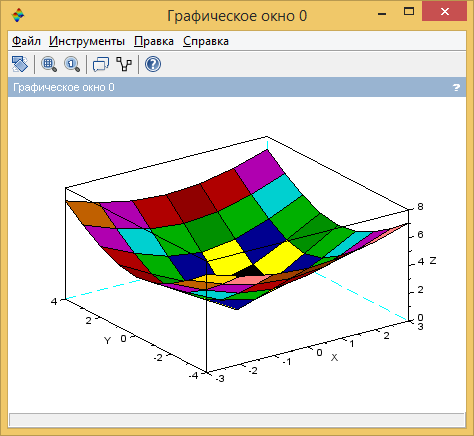


Рисунок 8 – График функции параболы

Для функции график представлен на рисунке 9.

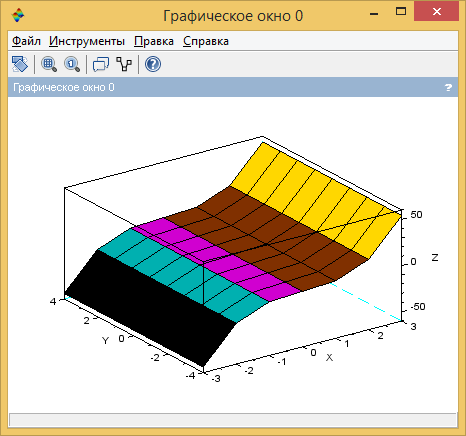


Рисунок 9 – График функции параболы

Задача

Построить график поверхности заданной следующей системой уравнений используя команду *eval3dp.*

Прежде всего определим массивы значений параметров p1 и p2. Далее создадим функцию *scp*, которая задает график.

Напомним, что функции в Scilab создаются при помощи команды *deff:*

*deff([s1,s2,...]=newfunction(e1,e2,...)*', где *s1,s2*,...список выходных переменных которым будет присвоен конечный результат вычислений;

*newfunction* – имя создаваемой функции оно будет использоваться для ее вызова входные параметры.

Обратите внимание что команда *deff* записана в три строки только для удобства чтения

Теперь сформируем прямоугольную сеть при помощи команды *eval3dp* и построим график обратившись к функции *plot3d.*

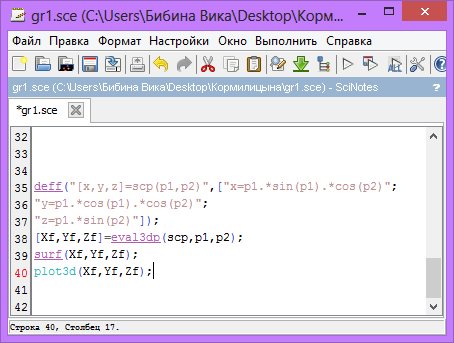


Рисунок 10– Код программы сферы

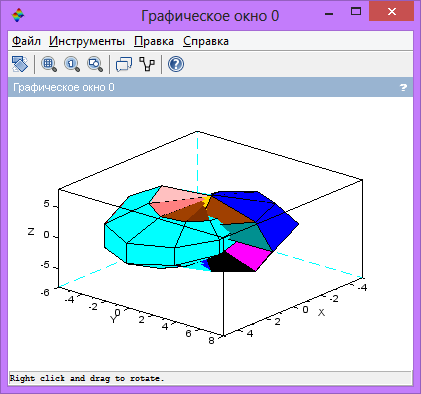


Рисунок 10– График системы

**Построение графиков поверхностей, заданных параметрически**

При построении графиков поверхностей, заданных параметрически x(u,v), y(u,v) и z(u,v) необходимо построить матрицы X, Y и Z одинакового размера. Для этого массивы u и v должны быть одинакового размера. После этого следует выделить два основных вида представления x, y и z в случае параметрического задания поверхностей:

1. Если x, y и z представимы в виде f(u)g(v), то соответствующие им матрицы X, Y и Z следует формировать в виде матричного умножения f(u) на g(v).
2. Если x, y и z представимы в виде f(u) или*.*g(v), то в этом случае матрицы X, Y и Z следует записывать в виде f(u)*.*ones(size(v)) или g(v).ones(size(u))
3. соответственно.

Рассмотрим задачу построения графика поверхности сферы

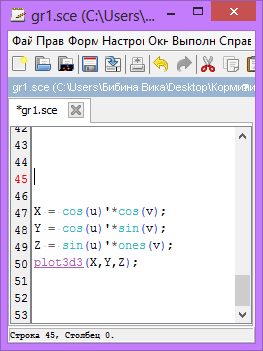


Рисунок 10– Код программы сферы

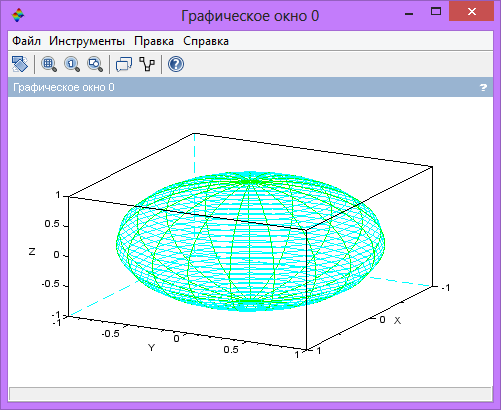
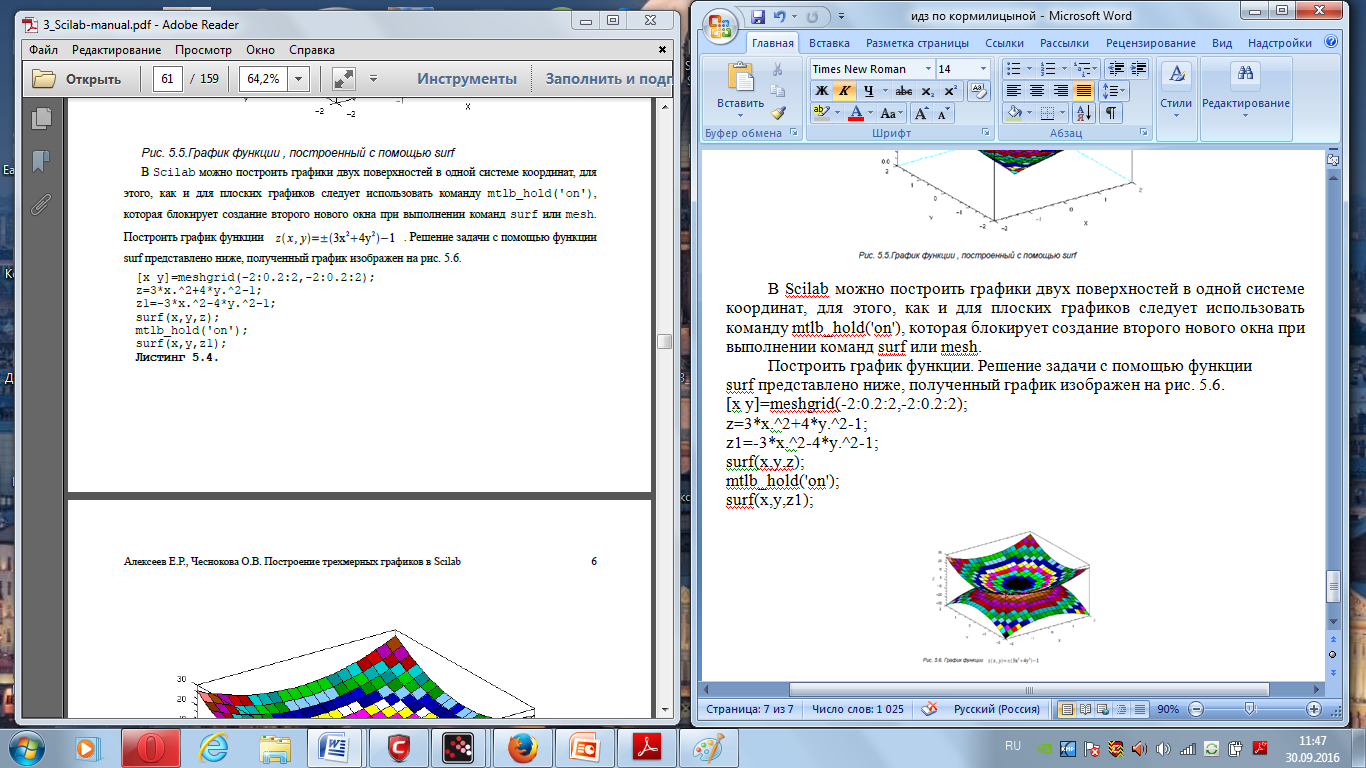


Рисунок 10– График сферы

В Scilab можно построить графики двух поверхностей в одной системе координат, для этого, как и для плоских графиков следует использовать команду mtlb\_hold('on'), которая блокирует создание второго нового окна при выполнении команд surf или mesh.

Построить график функции. Решение задачи с помощью функции surf:

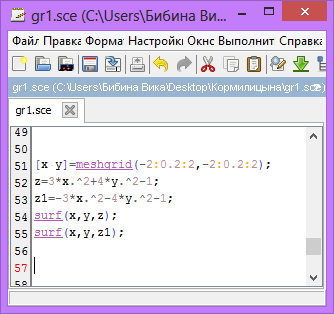


Рисунок 11 – Код программы

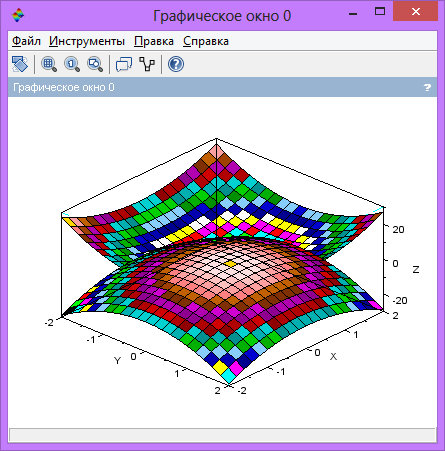


Рисунок 12– График функции

**Заключение**

Пакет прикладных математических программ Scilab, обладает большими возможностями для решения математических задач. В системе Scilab возможно решение примеров, выражений, а также операции над ними. Рассмотрен ряд функций при построении графиков функций: двумерных, трехмерных. При решении алгебраических уравнений в системе компьютерной алгебры Scilab используется функцияroots, а для построения графиков функцию plot.Для решения трансцендентных уравнений в Scilab применяется не только функцию roots, но и функцию , где начальное приближение, *f* функция, описывающая левую часть уравнения *.*для решения систем линейных уравнений применяется функция *linsolve*.

В данной работе «Решение математических задач в системе компьютерной математики Scilab» решены следующие задачи:

* познакомились с программным продуктомScilab, который позволяет решать различные математические задачи;
* изучили основные возможностипрограммы Scilab;
* разработали алгоритм решения нелинейных уравнений вScilab;
* разработали алгоритм решения систем линейных уравненийв Scilab;
* рассмотрелирешения математических задач в Scilab;

Итак, можно сделать вывод, что применение математического пакета Scilab удобно для решения математических задач, в частности – решения нелинейных уравнений и систем уравнений.

**Список использованных источников**

1. Алексеев, Е.Р. Scilab: Решение инженерных и математических задач / Е.Р. Алексеев, О.В. Чеснокова, Е.А. Рудченко.–М. :ALTlinux; БИНОМ. Лаборатория знаний, 2008. – 260с. : 8 с. Цв. Вклейки.–(Библиотека ALTLinux).
2. Аверьянов, Г.П. Современная информатика : учебное пособие / Г.П. Аверьянов, В.В. Дмитриева. - М. : МИФИ, 2011. - 436 с.
3. Акчурин, Э.А. Система компьютерной математики Scilab: учебное для студентов направления «Информатикам и вычислительная техника» / Э.А. Акчурин. – Самара : 2011. – 111 с.
4. Веселова, Л.В. Алгебра и теория чисел : учебное пособие / Л.В. Веселова, О.Е. Тихонов. - Казань : Издательство КНИТУ, 2014. - 107 с.
5. Галушкин, Н.Е. Высокоуровневые методы программирования: язык программирования MatLab : учебник / Н.Е. Галушкин. - Ростов-н/Д : Издательство Южного федерального университета, 2011. - Ч. 1. - 182 с.
6. Геворкян, П.С. Высшая математика. Основы математического анализа / П.С. Геворкян. - М. :Физматлит, 2007. - 238 с
7. Громов, Ю.Ю. Дискретная математика : учебное пособие / Ю.Ю. Громов, О.Г. Иванова, Ю.В. Кулаков и др. ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2012. - 128 с.
8. Плещинская, И.Е. Интерактивные системы Scilab, Matlab, Mathcad : учебное пособие / И.Е. Плещинская, А.Н. Титов, Е.Р. Бадертдинова, С.И. Дуев. - Казань : Издательство КНИТУ, 2014. - 195 с.
9. Гусева, Е.Н. Информатика : учебное пособие / Е.Н. Гусева, И.Ю. Ефимова, Р.И. Коробков и др. - 3-е изд., стереотип. - М. : Флинта, 2011. - 260 с.
10. Колокольникова, А.И. Спецразделы информатики: введение в MatLab : учебное пособие / А.И. Колокольникова, А.Г. Киренберг. - М. ; Берлин : Директ-Медиа, 2014. - 73 с.
11. Ремизов, А.О. Линейная алгебра и геометрия : учебное пособие / А.О. Ремизов, И.Р. Шафаревич. - М. :Физматлит, 2009. - 512 с.
12. Тропин, И.С.Численные и технические расчеты в среде Scilab(ПО для решения задач численных и техническихвычислений):учебное пособие / И.С. Тропин, О.И. Михайлова, А.В. Михайлов. - Москва : 2008. – 65 с.
13. Усачев, А.Е. Информатика : учебно-практическое пособие / А.Е. Усачев. - Ульяновск :УлГТУ, 2013. - 121 с.
14. Фомин, Д.В. Основы компьютерной электроники : учебное пособие / Д.В. Фомин. - М. ; Берлин : Директ-Медиа, 2014. - 108 с.