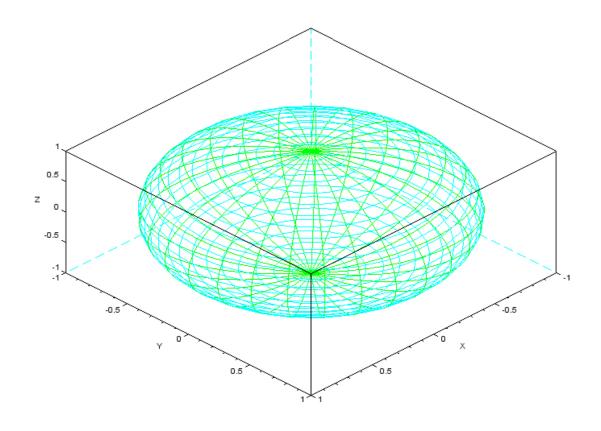


В. Ю. Захарова МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО КУРСУ «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ» ЧАСТЬ 1



Санкт-Петербург 2021

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

В.Ю. Захарова МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ЛАБОРАТОРНЫХ РАБОТ ПО КУРСУ "МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ФИЗИЧЕСКИХ ПРОЦЕССОВ". ЧАСТЬ 1

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО по направлению подготовки 14.03.01 Ядерная энергетика и теплофизика в качестве Учебно-методическое пособие для реализации основных профессиональных образовательных программ высшего образования бакалавриата



Санкт-Петербург 2021 Захарова В.Ю., Методические указания к выполнению лабораторных работ по курсу "Математическое моделирование физических процессов". Часть 1– СПб: Университет ИТМО, 2021. – 72 с.

Рецензент(ы):

Трифанова Екатерина Станиславовна, кандидат физико-математических наук, доцент, преподаватель (квалификационная категория "преподаватель") факультета систем управления и робототехники, Университета ИТМО.

Учебно-методическое пособие адресовано студентам бакалавриата, обучающимся по направлению 14.03.01 «Ядерная энергетика и теплофизика». В пособие входит описание двух лабораторных работ, направленных на освоение базовых принципов работы с математическим пакетом Scilab:

- 1. Основы работы в математическом пакете Scilab;
- 2. Визуализация результатов в математическом пакете Scilab. Работы снабжены пошаговыми примерами выполнения задач и описанием заданий для самостоятельной работы.

университет итмо

Университет ИТМО ведущий России вуз информационных и фотонных технологий, один из немногих российских вузов, получивших В 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

> © Университет ИТМО, 2021 © Захарова В.Ю., 2021

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ЛАБОРАТОРНАЯ РАБОТА №1. ОСНОВЫ РАБОТЫ В	
МАТЕМАТИЧЕСКОМ ПАКЕТЕ SCILAB	5
1 Основные сведения о работе	5
2 Общие сведения о математическом пакете Scilab	5
3 Интерфейс и основные команды пакета Scilab	10
4 Требования для получения допуска	29
5 Контрольное задание к лабораторной работе №1	30
ЛАБОРАТОРНАЯ РАБОТА №2. ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ В	
МАТЕМАТИЧЕСКОМ ПАКЕТЕ SCILAB	33
1 Основные сведения о работе	33
2 Особенности графической подсистемы пакета Scilab	34
3 Работа с графикой в пакете Scilab	38
4 Требования для получения допуска	64
5 Контрольное задание к Лабораторной работе № 2	65
СПИСОК ЛИТЕРАТУРЫ	72

ВВЕДЕНИЕ

Современное математическое моделирование физических процессов сложно представить без использования компьютерных средств, позволяющих автоматизировать решение уравнений математических моделей, обработку полученных в результате расчетов данных, построение графиков. В качестве таких средств выступают прикладные пакеты программ для математических расчетов. Среди подобных пакетов программ большой интерес представляет пакет Scilab, распространяющийся по лицензии GNU GPL, что означает его свободное копирование и использование без взимания дополнительной платы. Тем самым пакет доступен для студентов даже при работе на собственных компьютерах или ноутбуках. Пакет является многофункциональным, охватывающим различные типы математических моделей и методов расчета.

Настоящее пособие содержит описания двух лабораторных работ по курсу «Математическое моделирование физических процессов», предназначенных для обучающихся бакалавриата по направлению подготовки 14.03.01 «Ядерная энергетика и теплофизика». В процессе выполнения работ студенты получают представление о дескрипторной графике и матричном представлении данных в математическом программном обеспечении, учатся разрабатывать файлы сценария для пакета Scilab, получают навыки построения трехмерных графиков поверхностей и контурных графиков, проведения базовых математических операций, обработки массивов экспериментальных данных средствами пакета Scilab.

Каждая лабораторная работа состоит из теоретических сведений, пошагового практического примера, выполнение которого является обязательным, а также требований для получения допуска и контрольного задания по вариантам. Все примеры, рассматриваемые в пособии, относятся к математическому пакету Scilab версии 5.3.3 при работе в операционной системе Windows.

В пособии используются следующие обозначения.

Курсивом выделяются те действия, порядок выполнения которых уже был описан ранее. Эти действия подробно не поясняются, их необходимо выполнить самостоятельно.

Жирным шрифтом выделяются те положения, на которые необходимо обратить особое внимание.

Шрифтом Tahoma выделяются названия пунктов меню, переменных, параметров, а также те выражения, которые необходимо ввести с командной строки или в файл. Шрифтом Courier New выделяются сообщения в окне вывода результатов.

При выборе пунктов меню используется следующая сокращенная запись: «Пункт 1 > Подпункт 10», что означает, что необходимо вначале выбрать в меню Пункт 1, а затем из появившегося подменю выбрать Подпункт 10.

ЛАБОРАТОРНАЯ РАБОТА №1. ОСНОВЫ РАБОТЫ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ SCILAB

1 Основные сведения о работе

1.1 Цель работы

Целью работы является ознакомление с интерфейсом и основными функциями свободно распространяемого пакета научных программ для численных расчетов Scilab и получение навыков выполнения простейших расчетов.

1.2 Порядок выполнения работы

Работа состоит из двух частей. В первой части работы необходимо ознакомиться с теоретическим материалом, включающим общие сведения о математическом пакете Scilab, и выполнить практические примеры.

Для получения допуска ко второй части работы необходимо предъявить три файла, полученных в результате выполнения практических примеров, и ответить на теоретические вопросы.

Во второй части работы необходимо выполнить контрольное задание.

2 Общие сведения о математическом пакете Scilab

2.1 Назначение и возможности пакета

Пакет Scilab предназначен для применения в инженерных и научных расчетах. В первую очередь, он предназначен для автоматизации различных численных расчетов. При помощи этого пакета можно, в частности:

- Строить двухмерные и трехмерные графики;
- Создавать анимацию результатов расчетов;
- Решать задачи линейной алгебры, включая операции с разреженными матрицами;
- Производить интерполяцию и аппроксимацию экспериментальных данных и функций;
 - Решать обыкновенные дифференциальные уравнения;
 - Обрабатывать сигналы;
 - Решать статистические задачи;
 - Моделировать динамические системы.

Математический пакет Scilab имеет собственный язык программирования высокого уровня. Этот язык основан на языке, использующемся в математическом пакете MATLAB, хотя и отличается от него. Оба эти языка адаптированы для научных расчетов и являются объектно-ориентированными.

Математический пакет Scilab работает в режиме интерпретации, т.е. все команды сразу же выполняются. Это замедляет процесс вычислений по

сравнению с откомпилированными программами¹, но позволяет легко контролировать этот процесс.

2.2 Сравнение математических пакетов Scilab и MATLAB

Язык программирования в математическом пакете Scilab нельзя назвать идентичным языку программирования математического пакета MATLAB. Он сохраняет основные идеи языка математического пакета MATLAB, поскольку изначально строился на его основе. В то же время в математический пакет Scilab добавлены новые функции, а значительная часть функций математического пакета MATLAB отсутствует. Язык математического пакета Scilab больше приближен к языку программирования Си, тогда как язык математического пакета MATLAB исходно строился на библиотеках языка программирования Фортран. В этом заключается некоторое преимущество математического пакета Scilab.

Некоторые функции пакета Scilab заимствованы из языка программирования Си и различных его библиотек. Некоторые функции математических пакетов Scilab и MATLAB полностью совпадают. Т. е. они имеют одинаковый синтаксис и возвращают одни и те же результаты. В то же время часть функций пакета Scilab имеет отличный от пакета MATLAB синтаксис, но похожее действие.

Различие пакетов MATLAB и Scilab сильнее проявляется при решении сложных задач. Здесь, как правило, возможности пакета Scilab существенно меньше. В то же время в пакете Scilab устранены некоторые проблемы пакета MATLAB. Несмотря на широкий перечень решаемых задач, пакет Scilab на настоящий момент уступает коммерческому пакету MATLAB.

Для студентов, ранее работавших в пакете MATLAB, примеры в данных методических указаниях снабжены комментариями, указывающими на различие пакетов MATLAB и Scilab. Те студенты, которые не работали и не собираются работать в пакете MATLAB, могут пропустить подобные комментарии.

2.3 Scilab как некоммерческий пакет

Математический пакет Scilab является открытым программным продуктом, распространяемым по лицензии GNU GPL Лицензия GNU GPL, в целом, означает следующее:

- Любой пользователь может запустить программу с любой целью: образовательной, демонстрационной, коммерческой и пр.
- Любой пользователь может изучать, как работает программа или ее модификация. Для этого ему предоставляется доступ к исходному коду программы.
 - Можно свободно распространять копии программы.

_

¹ Такие программы, как правило, выполняются целиком после предварительного процесса компиляции.

– Можно свободно изменять и модифицировать программу, но только при условии, что новая программа тоже будет распространяться по лицензии GNU GPL или совместимой с ней.

большая Как часть открытого программного обеспечения, математический пакет Scilab исходно был ориентирован на операционную систему GNU/Linux. В настоящий момент широко распространена версия для операционной системы Microsoft Windows. Математический пакет Scilab входит состав некоторых дистрибутивов операционной системы Linux, ориентированных на решение научных задач, например, дистрибутива ALT Linux.

2.4 Особенности представления чисел в математическом пакете Scilab

Математический пакет Scilab, как и математический пакет MATLAB, работает не с отдельными числами, а с матрицами. В силу этого любое отдельное число рассматривается как матрица размера 1×1 . Такое представление чисел упрощает работу с массивами экспериментальных данных, решение систем уравнений и реализацию численных методов решения задач в целом.

Матрицы в математическом пакете Scilab можно рассматривать как массивы и использовать для целей программирования. Однако следует учитывать, что индексация массивов начинается не с нуля, как обычно принято в программировании, а с единицы. Это означает, что номер первого элемента массива — единица, а не ноль, как обычно принято. Такое представление удобнее при решении математических задач, так как в математике номер первого элемента матрицы — единица.

2.5 Арифметические операции и математические функции

Поскольку в математическом пакете Scilab все числа рассматриваются как матрицы, то все арифметические операции в нем можно рассматривать как операции над матрицами. Арифметические операции над матрицами проводятся по правилам соответствующих матричных операций, известных из математики.

Для матриц размера 1×1 (т. е. обычных чисел) определены операции, указанные в Табл. 1. Для изменения порядка операций можно использовать круглые скобки () по правилам, известным из математики.

Для матриц размера большего, чем 1×1 , необходимо помнить про ограничения, накладываемые на размер исходных матриц для каждой операции.

Все матричные операции в математическом пакете Scilab делятся на 2 группы:

- операции, выполняемые по правилам операций над матрицами;
- поэлементные операции.

Табл. 1 Арифметические операции над числами в математическом пакете Scilab

Обозначение	Название	Пример	Результат операции в примере
+	сложение	2+4	6
-	вычитание	2-4	-2
*	умножение	2*4	8
/	деление	2/4	0,5
^	возведение	2^4	16
·	в степень	Δ' 4	10

Поэлементные операции выполняют соответствующую арифметическую операцию над двумя числами, стоящими на одинаковых местах в матрицах, и помещают результат на то же место в матрице результата. Т.е. в этом случае для некоторой арифметической операции X над матрицами *а* и *b* верно следующее:

если c = aXb, то $c_{ij} = a_{ij}Xb_{ij}$, где a_{ij} , b_{ij} , c_{ij} — элементы соответствующих матриц, находящиеся на пересечении строки i и столбца j.

Поэлементные операции в пакете Scilab обозначаются так же, как и матричные операции, но начинаются со знака точка. Например, .* – поэлементное умножение. Поскольку операции сложения и вычитания по правилам операций над матрицами выполняются поэлементно, то они обозначаются одинаково во всех случаях, без точки.

В Табл. 2 приведены основные операции над матрицами и ограничения, накладываемые на размер матриц. Можно также производить операции сложения, умножения, деления и вычитания матрицы и числа. В этом случае соответствующая операция производится над каждым элементом матрицы.

В математический пакет Scilab встроены все часто использующиеся элементарные и специальные функции: тригонометрические функции, экспонента, логарифм, функции Бесселя различных порядков и пр. Их можно применять и к матрицам. В этом случае вычисление функции происходит поэлементно. Т. е., например, для матрицы $c = \sin(a)$ выполняется следующее соотношение $c_{ij} = \sin(a_{ij})$.

2.6 Типы переменных

Как и во многих других подобных пакетах программ, в пакете Scilab можно задавать собственные переменные и производить операции над ними. Существуют и встроенные переменные. Матрица (массив) действительных чисел — не единственный тип переменных, существующих в пакете Scilab. Существуют и строковые переменные, и структуры. Подробнее эти типы переменных будут рассмотрены в следующих лабораторных работах.

 Табл. 2 Основные операции над матрицами в математическом пакете

 Scilab

Символ и название операции	Ограничения	Пример	Порядок расчетов	Результат примера	
+ матричное сложение	матрицы должны иметь	$ \begin{pmatrix} 12 \\ 34 \\ 56 \end{pmatrix} + \begin{pmatrix} 33 \\ 34 \\ 44 \end{pmatrix} $	$ \begin{pmatrix} 1+32+3 \\ 3+34+4 \\ 5+46+4 \end{pmatrix} $	(4 5) 6 8 910)	
- вычитание матриц	одинаковый размер		$ \begin{pmatrix} 12 \\ 34 \\ 56 \end{pmatrix} - \begin{pmatrix} 33 \\ 34 \\ 44 \end{pmatrix} $	$ \begin{pmatrix} 1-32-3 \\ 3-34-4 \\ 5-46-4 \end{pmatrix} $	$\begin{pmatrix} -2-1\\0&0\\1&2 \end{pmatrix}$
* матричное умножение	число столбцов первой матрицы должно совпадать с числом строк второй матрицы	$ \begin{pmatrix} 12\\34\\56 \end{pmatrix} * \begin{pmatrix} 7 & 8\\3 & 4 \end{pmatrix} $	$ \begin{pmatrix} 1*7 + 2*3 & 1*8 + 2*4 \\ 3*7 + 4*3 & 3*8 + 4*4 \\ 5*7 + 6*3 & 5*8 + 6*4 \end{pmatrix} $	$ \begin{pmatrix} 13 & 16 \\ 33 & 40 \\ 43 & 54 \end{pmatrix} $	
.* поэлементное умножение	матрицы должны иметь одинаковый размер	$ \begin{pmatrix} 12 \\ 34 \\ 56 \end{pmatrix} * \begin{pmatrix} 33 \\ 34 \\ 44 \end{pmatrix} $	$ \begin{pmatrix} 1*32*3 \\ 3*34*4 \\ 5*46*4 \end{pmatrix} $	$ \begin{pmatrix} 3 & 6 \\ 9 & 16 \\ 20 & 24 \end{pmatrix} $	
./ поэлементное деление .^ поэлементное		$ \begin{pmatrix} 12 \\ 34 \\ 56 \end{pmatrix} \cdot \begin{pmatrix} 33 \\ 34 \\ 44 \end{pmatrix} $ $ \begin{pmatrix} 12 \\ 34 \\ 44 \end{pmatrix} \cdot \begin{pmatrix} 33 \\ 34 \\ 44 \end{pmatrix} $	$ \begin{pmatrix} 1/3 & 2/3 \\ 3/3 & 4/4 \\ 5/4 & 6/4 \end{pmatrix} $ $ \begin{pmatrix} 1^3 & 2^3 \\ 3^3 & 4^4 \end{pmatrix} $	$ \begin{pmatrix} 0.33 \ 0.67 \\ 1 & 1 \\ 1.25 & 1.5 \end{pmatrix} $ $ \begin{pmatrix} 1 & 8 \\ 9 & 256 \end{pmatrix} $	
возведение в степень		(56) (44)	$\left(5^4 6^4\right)$	(625 1236)	

В пакете Scilab существуют константы, которые могут быть встроенными или определенными пользователем. Константы защищены от записи или иного изменения их значения. Имена констант в пакете Scilab начинаются со знака процент %. Наиболее часто используемые встроенные константы приведены ниже:

%рі – число π ,

%і – мнимая единица,

%inf – машинная бесконечность,

%eps – точность расчетов с плавающей запятой 2 .

 $^{^2}$ В математическом пакете MATLAB таких констант нет.

2.7 Режимы работы

Пакет Scilab выполняет команды в режиме интерпретации, т. е. введенная команда сразу же выполняется. Пакет Scilab не компилирует программы. Команды можно вводить непосредственно в командную строку. Этот режим удобен для простых расчетов или знакомства с возможностями пакета Scilab. Отменить действие неправильно введенной в командную строку команды невозможно.

Непосредственный ввод команд неудобен при решении сложных задач. Ошибка при вводе команды может привести к необходимости повторного набора всех команд. В этом случае удобнее использовать так называемые файлы сценария³, которые представляют собой набор тех же команд, которые набираются в командной строке. Эти команды выполняются так же, как и из командной строки. Подобные файлы можно редактировать и использовать многократно.

В отличие от командной строки, файл сценария выполняется только после загрузки в среду Scilab, и сразу целиком. Загрузка выполняется специальной функцией.

3 Интерфейс и основные команды пакета Scilab

3.1 Запуск приложения и первое знакомство

- 3.1.1. Запустите Scilab, выбрав иконку на рабочем столе или соответствующий пункт в меню Пуск операционной системы Windows⁴.
- 3.1.2. Появится командное окно среды Scilab. В зависимости от версии операционной системы оно будет называться либо Командное окно (Scilab), либо Console. В командном окне видно приглашение командной строки → (Рис. 1.1). В командную строку пользователем вводятся команды пакета Scilab. Результаты вычислений, как правило, также выводятся в командном окне.
- 3.1.3. Чтобы зайти в помощь по программе, нажмите на пункт Справка⁵ в главном меню.
- 3.1.4. Выберите пункт Содержание из раскрывшегося в результате меню. Здесь можно просмотреть справку по интересующей команде. К сожалению, справка приводится только на английском языке. Закройте окно со справкой, запомнив ее расположение.

³ В английском варианте эти файлы называются script. Однако в русскоязычной литературе стало распространенным название «файл сценария», хотя название «командный файл» ближе по смыслу.

⁴ Возможна работа и в операционной системе Linux, для запуска используйте соответствующие команды.

 $^{^{5}}$ В некоторых версиях программы название пункта справка заменено на знак вопроса ?

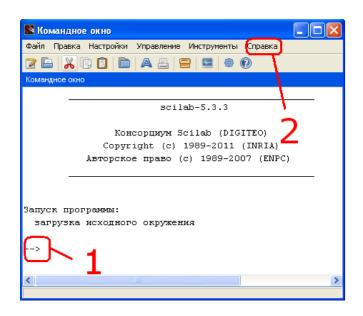


Рис 1.1 – Командное окно среды Scilab. 1 – приглашение командной строки, 2 – справка.

3.1.5. По желанию просмотрите демонстрационные примеры, выбрав из главного меню пункт Справка > Примеры. Также примеры можно запустить, выбрав иконку на панели инструментов. Рекомендуется просмотреть хотя бы некоторые примеры из категории Графика (или Graphics). Эти примеры демонстрируют графические возможности пакета Scilab. Закройте окно Примеры после просмотра примеров.

3.2 Работа в режиме калькулятора

3.2.1. Пакет Scilab очень просто использовать в режиме калькулятора⁶. Просто наберите нужное выражение в командной строке. Например, для вычисления произведения 3·8 введите (рис. 1.2).

Нажмите клавишу ENTER, чтобы вычислить введенное выражение.

- 3.2.2. Если результат введенного выражения не был присвоен ни одной переменной, то он автоматически присваивается программой переменной ans, как и произошло в рассматриваемом случае. Все переменные, значения которых изменялись, сразу же выводятся в командном окне. В результате сразу виден результат вычисления выражения 3*8=24.
- 3.2.3. Для вычисления арифметических выражений используются стандартные обозначения, указанные в п. 0. Порядок выполнения операций

⁷ Далее в тексте везде знак \rightarrow стоит перед выражениями, которые необходимо ввести в командном окне. Сам знак вводить не нужно.

⁶ В этом режиме работа с пакетом Scilab мало отличается от работы с пакетом MATLAB, отличия будут специально оговорены.

такой же, как и в обычных арифметических выражениях. Например, выражение $\frac{10,5-13}{1-1}$ (10-3) вычисляется следующим образом:

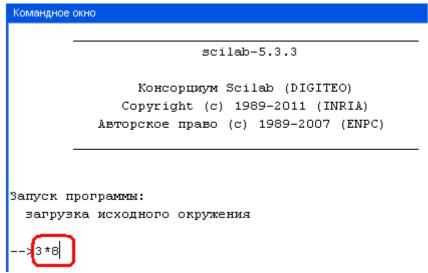


Рис 1.2 – Вычисление произведения в командном окне Scilab

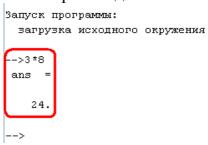


Рис 1.3 – Результат вычисления в командном окне Scilab

3.2.4. Переменную ans также можно использовать в выражениях. Однако, не следует забывать, что ее значение меняется после каждой операции. Например, выражение $\frac{(5.5+6.1-1)\cdot(5.5+6.1+5)}{(5.5+6.1-1)\cdot(5.5+6.1+5)-1}$ можно вычислить, не повторяя несколько раз одни и те же операции.

$$\rightarrow$$
 5.5+6.1 ans⁹ = 11.6¹⁰

 8 Результаты вычислений, появляющиеся в командном окне, приводятся в методических указаниях без знака \rightarrow и выделяются шрифтом Courier New. Лишние пробелы далее удаляются для экономии места.

⁹ Для уменьшения объема методических указаний, в них иногда не приводятся пробелы и знаки перевода строки в результатах, выводимых пакетом Scilab.

¹⁰ В отличие от пакета MATLAB, в пакете Scilab по умолчанию не выводятся последние нули после запятой. Количество выводимых знаков выбирается в зависимости от результата.

```
\rightarrow (ans-1) * (ans+5)
ans = 175.96
\rightarrow ans/(ans-1)
ans = 1.0057156
```

Если набирать выражение целиком, то результат будет такой же. Проверьте.

```
\rightarrow (5.5+6.1-1)*(5.5+6.1+5)/((5.5+6.1-1)*(5.5+6.1+5)-1) ans = 1.0057156
```

- 3.2.5. Во многих случаях не нужно выводить результат промежуточных вычислений на экран, особенно, если он представляет собой матрицу больших размеров. В этом случае после выражения, результат которого не нужно выводить в командное окно, ставится знак точка с запятой; . Обратите внимание на поведение среды Scilab в следующем примере.
 - → 15*25;
 → ans/5ans = 75.
- 3.2.6. Прописные и строчные буквы в среде Scilab различаются, поэтому если ввести название переменной ans с прописной буквы, то выведется сообщение об ошибке.
 - →3*Ans !--error 4 Неизвестная переменная: Ans
- 3.2.7. Если появилось сообщение об ошибке, то исправить ее можно, просто набрав правильную команду. Если большая часть команды была ранее набрана правильно, то можно ее отредактировать, нажав клавишу вверх ↑ на клавиатуре. Нажмите клавишу ↑ и используйте стрелки для редактирования ранее введенной неправильной команды.
 - \rightarrow 3*ans
- 3.2.8. Сохраняется не только последняя введенная команда, а все команды в текущем сеансе работы. Получить к ним доступ можно, нажимая клавишу ↑ несколько раз перед введением новой команды. Можно также использовать клавишу ↓, если нужная команда была случайно пропущена при нажатии клавиши ↑. Найдите команду, задающую ранее введенное в п. 3.2.4. арифметическое выражение, и замените в нем последнюю цифру 1 на 1.6. Нажмите клавишу ENTER, чтобы вычислить новое выражение.

```
\rightarrow (5.5+6.1-1)*(5.5+6.1+5)/((5.5+6.1-1)*(5.5+6.1+5)-1.6) ans = 1.0091764
```

3.2.9. Нажатие клавиши ENTER приводит к немедленному выполнению введенной команды, если она не содержит операторов цикла и ветвления или многоточия Многоточие используется, чтобы вводить длинные выражения, занимающие несколько строк. Если строка оканчивается многоточием, то при нажатии клавиши ENTER происходит переход на другую строку. В новой строке

необходимо продолжить выражение, не вводя никаких дополнительных символов.

Выражение в пункте 3.2.8. можно вычислить и следующим образом.

- \rightarrow (5.5+6.1-1) * (5.5+6.1+5) / ((5.5...
- → +6.1-1) * (5.5+6.1+5) -1.6)

ans = 1.0091764

3.2.10. В выражения могут входить тригонометрические функции. Аргумент функций указывается в скобках. Все угловые меры задаются в радианах. Вычислите значение выражения $\sin(2\pi)\cdot\cos\left(\frac{\pi}{3}\right)$, как описано ниже.

Число π является в пакете Scilab константой и обозначается %pi¹¹. Все имена констант начинаются со знака %¹². Они защищены от записи, их значение не может быть изменено.

```
\Rightarrowsin(2*%pi)*cos(%pi/3)
ans = -1.225D-16<sup>13</sup>
```

3.2.11. Список тригонометрических и других элементарных функций можно найти в помощи, раздел Elementary Functions.

Наиболее часто употребляемые функции приведены в Табл. 3.¹⁴

3.2.12. В целом работа со средой Scilab в режиме калькулятора напоминает работу двухстрочного калькулятора и обычно не представляет особых затруднений.

3.3 Особенности представления чисел в пакете Scilab

3.3.1. В пакете Scilab существует особое число — машинная бесконечность, которая обозначается как $\%inf^{15}$. Операции над бесконечностью проводятся по известным из математики правилам.

```
→5+%inf
ans = Inf<sup>16</sup>
→%inf*(-6)
ans = -Inf
```

3.3.2. Из математики известно, что некоторые операции над бесконечно большими или бесконечно малыми величинами приводят к неопределенности.

```
→%inf/%inf
ans = Nan
→%inf*0
ans = Nan
```

¹¹ В пакете MATLAB число π – функция без аргументов, обозначается рі.

¹² В пакете MATLAB такого типа переменных нет.

¹³ Точное значение приведенного выражения – ноль, но необходимо помнить про погрешность численных расчетов.

¹⁴ Приведенные в таблице функции совпадают с функциями пакета MATLAB.

¹⁵ В пакете MATLAB бесконечность обозначается просто Inf

¹⁶ %inf – это константа, а Inf – ее значение.

Значение Nan является аббревиатурой от словосочетания Not a Number 17 . Любые арифметические операции над переменной, чье значение равно Nan, не могут изменить это значение.

Табл. 3 Основные элементарные и тригонометрические функции в пакете **Scilab**

T	ригонометрические функции	прочие функции		
sin	синус, sin	exp	экспонента, е ^х	
cos	косинус, cos	log	натуральный логарифм, ln	
tan	тангенс, tg	log10	десятичный логарифм, log	
asin	арксинус, arcsin	sqrt	квадратный корень, √	
sinh	гиперболический	abs	абсолютное значение,	
	синус, sh		модуль,	

3.3.3. Вопреки ожиданиям, деление на нуль не приводит к получению результата, равного бесконечно большой величине¹⁸. Вместо этого выводится сообщение об ошибке.

!-error 27

Деление на нуль...

3.3.4. При вычислении квадратного корня из отрицательного числа ошибка не возникает.

$$\rightarrow$$
sqrt(-9) ans = 3.i

Пакет Scilab работает с комплексными числами. Мнимая единица обозначается %i, т. е. это встроенная константа. При выводе результатов в командном окне знак % опускается. Комплексные числа записываются в виде a+b*%i, где a- действительная часть комплексного числа, b- мнимая.

3.3.5. Арифметические операции над комплексными числами выполняются подобно операциям над вещественными числами.

$$\rightarrow$$
 (5+3*%i)*(6+5*%i)-(25+2*%i)^2 ans = -606. - 57.i

- 3.3.6. Основные функции для работы с комплексными числами приведены ниже:
 - abs модуль комплексного числа,
 - imag мнимая часть комплексного числа,
 - real действительная часть комплексного числа,
 - сопј комплексно-сопряженное число.

_

¹⁷ Нечисловой характер данных (англ.)

¹⁸ В пакете MATLAB при делении на ноль получается бесконечность, а сообщение об ошибке не выводится. Выводится только предупреждение, которое не приводит к остановке расчетов. Обычно это неудобно.

```
→real(25+8*%i)
           25.
ans
```

3.3.7. Точность расчетов с плавающей точкой определяется встроенной константой %ерѕ

```
→%eps
 %eps
            2.220D-16
```

3.3.8. Числа, превышающие максимально допустимое положительное число, автоматически преобразуются в бесконечность. Числа, меньшие минимально допустимого положительного числа, автоматически преобразуются в ноль.

```
→1.23e1000
ans
    =
         Tnf
→1.23e-1000
ans = 0.
→-1.4e1000
    = -Tnf
ans
```

3.3.9. Формат вывода результатов не влияет на результаты расчетов. Все расчеты проводятся с точностью, указанной в константе %ерs (п. 3.3.7). По умолчанию выводятся не более восьми знаков после запятой. Если крайние правые цифры, стоящие после запятой, являются нулями, то они не выводятся. При необходимости автоматически происходит переход к экспоненциальному ϕ ормату¹⁹.

Для изменения формата вывода всех последующих результатов используется функция format. Ее первый аргумент – вид формата: V' – с (при необходимости плавающей запятой происходит переход экспоненциальному формату), 'e' – экспоненциальный. Второй аргумент – количество символов при выводе, считая символ знака числа и точку разделитель целой и дробной части.

Формат по умолчанию

```
→5.81796584
           5.8179658
 ans
```

Пять знаков, т. е. в данном случае два знака после запятой.

```
\rightarrow format('v',5)
→5.81796584
 ans
              5.82
```

```
3.3.10. Экспоненциальный формат
→format('e',5)
→5.81796584
             5.8D+00
3.3.11. Возврат к формату по умолчанию
→format('v')
```

¹⁹ Пакет MATLAB использует другой формат вывода результатов. Функция **format** в пакете MATLAB имеет другой синтаксис и работает по-другому

$$\rightarrow$$
 256e2 ans = 25600.

3.4 Помощь из командной строки

3.4.1. К справке по любой функции пакета Scilab можно быстро перейти, набрав в командной строке имя функции, по которой необходимо получить справку, после команды help.

Например, получите помощь по функции sin (синус)

$$\rightarrow$$
 help sin

В окне справки выводится соответствующий раздел 20 .

3.4.2. Просмотрите помощь по функции besseli — вычисление функции Бесселя различных порядков²¹. Закройте окно Help Browser (Справка).

3.5 Работа с переменными

В пакете Scilab тип новой создаваемой переменной определяется программой автоматически и соответствует типу первого присвоенного ей значения. В том числе, автоматически определяется размерность массива. Размерность массива в дальнейшем может быть изменена.

В программах для пакета Scilab желательно инициализировать переменные, т. е. сразу присваивать им начальные значения соответствующего типа, например, присваивать нуль для чисел. Иначе возможно неправильное определение типа переменной программой. Особенно часто ошибки возникают при автоматическом определении размера массива.

Строчные и прописные буквы в имени переменной различаются, т.е. переменные CV и CV – различные.

3.5.1. Значение 5 переменной х присваивается без предварительного объявления типа переменной.

3.5.2. После знака присваивания = может стоять не число, а выражение $\rightarrow_{y=2*sin}(x)$

$$y = -1.91785$$

3.5.3. Если нет необходимости в выводе переменной на экран, то в конце строки ставится точка с запятой ;.

→
$$z=-5*(x+y);$$

3.5.4. Для вывода значения любой переменной достаточно просто ввести ее имя

$$\Rightarrow_{x}$$
 $x = 5.$

²⁰ В пакете MATLAB в этом случае справка выведется в командном окне.

²¹ Курсивом в методических указаниях обычно выделяются действия, которые необходимо выполнить самостоятельно.

$$\rightarrow$$
z z = -15.4108

3.5.5. Просмотреть список определенных переменных можно командой who

 \rightarrow who

Обратите внимание, что помимо определенных нами ранее переменных x, y, z отображаются и встроенные переменные, и константы. Такие как %inf, %i и πp^{22} .

3.5.6. Посмотреть подробную информацию о переменных можно командой whos

 \rightarrow whos

В старых версиях программы, если список переменных не помещается на экран, то нажмите любую клавишу для продолжения вывода или клавишу **п**, чтобы прервать вывод. В новых версиях выведенный список можно пролистать при помощи полосы прокрутки окна.

3.5.7. Обычно информация обо всех переменных не нужна. Можно просмотреть информацию только о переменных, название которых начинается с какой-то определенной комбинации символов. Например, чтобы просмотреть информацию о константах (они, как известно, начинаются с %)²³ наберите

→whos -name %

3.5.8. Просмотрите информацию о переменной у

 \rightarrow whos -name y

Name Type Size Bytes y constant 1 by 1 24

Здесь Name — имя переменной, Type — тип переменной 24 , Bytes — количество байт, занимаемое переменной в памяти. В столбце Size отображается размерность массива. Как отмечено ранее, пакет Scilab рассматривает все числовые переменные как массивы. Одиночное число представляется массивом с размерностью 1×1 .

3.5.9. Ненужные переменные необходимо удалять из памяти, чтобы не занимать лишнее место. Особенно это важно при расчетах, требующих значительного объема оперативной памяти. Удаление переменной **х** из памяти:

 \rightarrow clear x

Проверим, удалилась ли переменная

 \rightarrow whos -name x

Name Type Size Bytes xcoslib library 928

 $^{^{22}}$ Пакет MATLAB при вызове команды who выдает только список переменных, определенных пользователем.

²³ В пакете MATLAB команда whos имеет другой синтаксис.

²⁴ В целом типы переменных в пакетах MATLAB и Scilab отличаются, хотя обычно они совместимы.

Переменной X среди перечисленных переменных нет²⁵.

3.5.10. Удаление всех переменных пользователя из памяти, встроенные переменные не удаляются

→ clear

Проверка, все ли удалено

→ who

Обратите внимание, что встроенные переменные не удалились.

3.6 Работа с массивами²⁶

В пакете Scilab все числовые переменные представляются в форме матриц. С точки зрения программирования матрица – это массив.

Число представляется как массив (матрица) с размерностью 1×1.

3.6.1. Поскольку все числовые переменные по умолчанию являются массивами, то для ввода массивов не нужно указывать никаких специальных параметров.

Например, вектор-столбец $a = \begin{pmatrix} 1 \\ 8 \\ 9 \end{pmatrix}$ задается следующим образом:

элементы вектора заключаются в квадратные скобки [], а строки разделяются точкой с запятой ;.

3.6.2. Столбцы в вектор-строке разделяются пробелом или запятой ,. Введите вектор-строку b=(1 8 9).

$$\Rightarrow$$
 b=[1 8 9]
b = 1. 8. 9.

Или, что равнозначно

$$\Rightarrow$$
 b=[1,8,9]
b = 1. 8. 9.

3.6.3. Матрицы вводятся построчно, начиная с верхней строки. Строки матрицы разделяются, как и в векторах, точкой с запятой, а элементы в строке

отделяются знаками пробел или запятая. Матрица $c = \begin{pmatrix} 1 & 5 & 6 \\ 4 & 8 & 9 \\ 9 & 0 & 2 \end{pmatrix}$ вводится

следующим образом:

²⁵ В зависимости от установленных и загруженных библиотек, пакетов расширения и версии пакета Scilab, могут отражаться различные переменные или не отражаться ничего.

²⁶ Указанные в данном пункте действия необходимо выполнить, не выходя из среды Scilab. В противном случае, некоторые действия будет необходимо выполнить заново.

3.6.4. Существует другой способ ввода матриц. Вместо точки с запятой; после каждой строки матрицы нужно нажимать клавишу ENTER. Этот способ ввода неудобен для использования в программах, хотя легко применим в командной строке.

```
⇒ c=[1 5 6

⇒4 8 9

⇒9 0 2]

c = 1. 5. 6.

4. 8. 9.

9. 0. 2.
```

3.6.5. Сцепляются (в единую матрицу) матрицы и вектора просто. Их необходимо указать на соответствующей позиции в новой матрице. Однако будьте внимательны с размерностью. Новая матрица должна иметь одинаковое количество элементов в каждом столбце и каждой строке.

Объединение двух вектор-столбцов v1 и v2 в один столбец

```
→ v1=[1;2];
→ v2=[3;4;5];
→ v=[v1;v2]
v = 1.
2.
3.
4.
5.
```

Неправильное объединение двух вектор-столбцов в матрицу

Несогласованное количество столбцов или строк.

3.6.6. Объединение двух вектор-столбцов в одну матрицу

→
$$v3=[6;7;8];$$
→ $M=[v2 \ v3]$
 $M = 3. 6.$
 $4. 7.$
 $5. 8.$

3.6.7. Чтобы обратиться (или использовать в выражении) к одному элементу вектора, а не ко всему вектору, нужно указать номер этого элемента в круглых скобках после имени вектора.

ИНДЕКСАЦИЯ МАССИВОВ ПО УМОЛЧАНИЮ НАЧИНАЕТСЯ С ЦИФРЫ 1, А НЕ 0, КАК ОБЫЧНО ПРИНЯТО В ЯЗЫКАХ ПРОГРАММИРОВАНИЯ. т. е. первый элемент массива имеет номер 1.

Присваивание переменной а1 первого элемента вектора **Vec**

```
→ vec=[1 4 7 9 10 12 15];
→ a1=vec(1)
a1 = 1.
```

3.6.8. Строка, состоящая из третьего, четвертого и первого элемента вектора **Vec** в указанном порядке.

```
\rightarrow a3=vec([3 4 1])
a3 = 7. 9. 1.
```

3.6.9. Строка, состоящая из элементов вектора Vec с третьего по шестой по порядку

```
\rightarrow a4=vec(3:6)
a4 = 7. 9. 10. 12.
```

3.6.10. Чтобы обратиться к отдельному элементу матрицы, необходимо указать номер строки и номер столбца. Третий элемент в первой строке матрицы m:

$$\rightarrow$$
 m=[1 2 4;5 6 7]
m = 1. 2. 4.
5. 6. 7.
 \rightarrow am=m(1,3)
am = 4.

3.6.11. Если при указании номера элемента двумерного массива указать только одну цифру, то сообщения об ошибке не выводится. Выведется тот элемент, который имеет соответствующий номер в компьютерном представлении матрицы. Это может быть удобно, если необходимо перебрать все элементы матрицы по порядку.

```
→ am5=m(5)
am5 = 4.
3.6.12. Из матрицы можно выделить блок
→ ab=m(1:2, 2:3)
ab = 2. 4.
6. 7.
```

3.6.13. Размерность массива можно узнать функцией **ndims**. В математике размерность 0 соответствует одиночному числу, 1 – вектору, 2 – двумерному массиву. Пакет Scilab во всех этих случаях показывает размерность 2, так как все числа в нем представляются как матрицы.

```
\rightarrow ndims(vec)
ans = 2.
\rightarrow ndims(m)
ans = 2.
```

3.6.14. Размер массива возвращает функция **SiZe**. Она возвращает вектор, состоящий из двух значений. Первое значение — количество строк массива, второе — количество столбцов массива.

```
\rightarrow size(vec) ans = 1. 7.
```

Если необходимо присвоить результат, возвращаемый функцией **size**, каким-то переменным, то можно поступить по-разному:

$$\Rightarrow$$
 z=size(m)
z = 2. 3.

Здесь z – вектор-строка, состоящая из двух чисел.

$$\rightarrow$$
 [z1 z2]=size(m)

$$z2 = 3.$$

$$Z1 = 2.$$

Здесь результат, возвращаемый функцией, разбит на две переменные.

Такой порядок действий допустим для большинства функций, возвращающих несколько значений.

3.6.15. Для вычисления длины вектора вместо функции size проще использовать функцию length:

$$\rightarrow$$
 length (vec) ans = 7.

3.6.16. Можно присваивать значение не всей матрице сразу, а каждому ее отдельному элементу.

$$\rightarrow$$
 vec(4)=8 vec = 1. 4. 7. **8.** 10. 12. 15.

3.6.17. Для присваивания значения отдельному элементу матрицы желательно, чтобы матрица уже существовала. В противном случае пакет Scilab может выбрать неправильный размер матрицы. Например, надо присвоить значение 5 второму элементу вектора nr, состоящему из пяти элементов. Можно просто набрать

$$\rightarrow \text{ nr } (2) = 5$$
nr = 0.
5.

Заданный вектор будет состоять из двух элементов.

3.6.18. Для быстрого задания векторов и матриц существует ряд функций, которые возвращают матрицу определенного типа. Для создания вектора с числами, образующими арифметическую прогрессию с заданным шагом, используется оператор двоеточие: (он ранее использовался в 3.6.12 для выделения блока из матрицы). Задается начальное значение, шаг и конечное значение, которые отделяются двоеточием. Шаг может быть отрицательным, тогда числа в последовательности будут убывать.

$$\Rightarrow$$
 x=1:5:60
x = 1. 6. 11. 16. 21. 26. 31. 36. 41. 46. 51. 56.²⁷

3.6.19. Если шаг не задан, то он полагается равным единице 1.

$$\rightarrow$$
 x=-1:6
x = -1. 0. 1. 2. 3. 4. 5. 6

3.6.20. Задание матрицы заданного размера (3×6), заполненной нулями:

 $^{^{27}}$ Последнее значение в последовательности не обязательно равно конечному значению. Оно не должно превышать его. Это зависит от шага.

3.6.21. Единичная матрица²⁹ заданного размера (4×4):

```
\rightarrow I=eye(4,4)
T =
         1.
                0.
         0.
                1.
                       0.
                              0.
         0.
                0.
                       1.
                              0.
                0.
         0.
                       0.
                              1.
```

3.6.22. Матрица, заполненная единицами (не путать с единичной матрицей):

```
\rightarrow E=ones (3,8)
                  1. 1.
F. =
        1.
                                  1.
                  1.
                       1.
                             1.
             1.
                                  1.
                                             1.
             1.
                  1.
                       1.
                            1.
                                  1.
```

3.6.23. Матрица, заполненная случайными числами, с равномерным распределением на интервале от нуля до единицы:

```
\rightarrow R=rand(3,5)
```

3.6.24. Прямоугольная матрица, заполненная случайными числами, распределенными по нормальному закону:

$$\rightarrow$$
RN=rand(3,5,'normal')³⁰

3.7 Особенности операций над матрицами³¹

3.7.1. Поскольку массивы в пакете Scilab воспринимаются как матрицы, то арифметические операции над матрицами в выражениях проводятся по правилам матричного сложения, вычитания умножения и деления. При этом матрицы должны иметь соответствующую размерность.

Для сложения и вычитания, как известно из математики, матрицы должны иметь одинаковую размерность:

```
→ matr1= [1 2; 8 9];
→ matr2= [3 4; 1 1];
→ matr3= [1 4 22];
→ matr1+matr2
ans = 4. 6.
9. 10.
```

 $^{^{28}}$ В отличие от пакета MATLAB функция zeros(3) в пакете Scilab не выдаст квадратную матрицу размера 3x3.

²⁹ В единичной матрице все элементы равны нулю, кроме находящихся на главной диагонали. Элементы на главной диагонали равны единице.

³⁰ В пакете MATLAB то же самое делает другая функция RN=randn(3,5)

³¹ Указанные в данном пункте действия необходимо выполнить, не выходя из среды Scilab. В противном случае, некоторые действия будет необходимо выполнить заново.

Некорректное суммирование.

3.7.2. Как и в математике, матрицы можно умножать и делить на число.

⇒
$$5*matr1$$

ans = 5. 10.
40. 45.
⇒ $matr2/2$
ans = 1.5 2.
0.5 0.5

3.7.3. При умножении матриц их размеры должны соответствовать друг другу по правилам матричного умножения (число столбцов первой матрицы должно быть равно числу строк второй матрицы). Результат умножения матриц — скалярное произведение строк первой матрицы на столбцы второй матрицы, стоящее на соответствующих местах.

```
→ matr4=[1 3 5 ; 0.5 4 4.5];
→ matr5=[5 6 ; 8 9 ; 8 16];
→ matr4*matr5
ans = 69. 113.
70.5 111.
```

- 3.7.4. Операции деления матриц в математике нет, но в пакете Scilab деление обратно умножению матриц. Следовательно, деление соответствует операции решения системы алгебраических уравнений. Эта операция подробнее будет рассмотрена в последующих лабораторных работах.
- 3.7.5. Возведение матрицы в степень (-1) соответствует ее обращению (поиску обратной матрицы). Эта операция возможна только для квадратных матриц.

$$\rightarrow$$
 matr2^(-1)
ans = -1. 4.
1. -3.

3.7.6. Операции сложения и вычитания по правилам математики выполняются над матрицами поэлементно (см. п. 0). Чтобы проводить поэлементно умножение, деление и возведение в степень для матриц, необходимо использовать операции .*, ./ и .^ вместо операций *, / и ^ соответственно.

Сравните результаты следующих вычислений:

3.7.7. Остальные математические функции, включая элементарные и специальные функции, как правило, работают над матрицами поэлементно.

Например, поэлементно работают тригонометрические функции.

3.7.8. Благодаря наличию поэлементных операций над матрицами, можно одной командой проводить вычисления для совокупности чисел по одной и той же формуле.

Например, необходимо вычислить значение выражения $3 \cdot e^{-x} \cdot \cos(x) + 5$ для аргументов x = 5, 8, 9, 12. Можно, конечно, вычислить выражение для каждого из чисел отдельно. А можно сделать проще:

$$\Rightarrow$$
x=[5 8 9 12];
 \Rightarrow y=3*exp(-x).*cos(x)+5
y = 5.00573 4.99985 4.99966 5.00002

Обратите внимание, что при вычислениях используется операция поэлементного умножения.

3.7.9. Ряд функций выполняет операции не над отдельными числами в матрице, а над всей матрицей в целом. Это, как правило, векторные операции.

Транспонирование матрицы:

3.7.10. Сумма всех элементов вектора или матрицы:

$$\rightarrow$$
v1=[1 2 3 4 5];
 \rightarrow sum(v1)
ans = 15.
 \rightarrow M=[3 8 23; 5 9 56; 5 9 7]
M = 3. 8. 23.
5. 9. 56.
5. 9. 7.

```
\rightarrowsum (M) 32
                125.
 ans
3.7.11. Сумма элементов каждого столбца матрицы:
\rightarrowsum (M, 'r')
                13.
                          26.
                                     86.
 ans
3.7.12. Сумма элементов каждой строки матрицы:
\rightarrowsum (M, 'c')
                34.
 ans
        =
                70.
                21.
```

3.7.13. Разверните командное окно Scilab во весь экран. Сохраните образ экрана компьютера с командным окном Scilab в графическом файле и предъявите файл преподавателю для получения допуска к выполнению лабораторной работы. Для сохранения образа экрана компьютера можно, например, поступить следующим образом. Нажмите клавиши Ctrl+Print Screen, откройте встроенный графический редактор, вставьте скопированный в буфер обмена графический образ, выбрав Правка > Вставить из меню редактора. Сохраните файл. Возможно использование инструментов «Набросок на фрагменте экрана» или «Ножницы», которые зависят от используемой операционной системы.

3.7.14. Сотрите из памяти все ранее определенные переменные:

→ clear

3.8 Файлы сценария

При проведении сложных вычислений использование командной строки неудобно по нескольким причинам. В частности, в случае обнаружения ошибки в вычислениях необходимо вводить команды заново, так как отмена предыдущей операции невозможна. Также необходимо вводить команды заново даже при решении нескольких однотипных задач. В командной строке неудобно использовать циклы и операции ветвления, а также другие средства программирования.

Использование файлов сценария позволяет решить описанные выше проблемы. Файл сценария состоит из последовательности команд, вводимых в том же виде, что и в командной строке. Команды выполняются последовательно, если в файле не встречаются команды, изменяющие порядок их выполнения.

Файл сценария и командная строка дополняют друг друга. Все переменные, определенные из файла сценария, в дальнейшем доступны из командной строки, и наоборот.

_

 $^{^{32}}$ В пакете MATLAB функция sum в применении к матрице выводит вектор, состоящий из чисел, равных сумме элементов соответствующего столбца матрицы. Для вычисления суммы всех элементов матрицы в этом случае необходимо использовать конструкцию sum(sum(M))

Файлы сценария в пакете Scilab имеют расширение *.SCe³³, но можно загрузить файл с любым другим расширением, содержащий команды пакета Scilab. Существуют и другие типы файлов пакета Scilab, которые будут рассмотрены в последующих работах.

- 3.8.1. Для создания файла сценария можно использовать любой текстовый редактор. Однако удобнее использовать встроенный текстовый редактор пакета Scilab³⁴ SciNotes. Чтобы запустить редактор, выберите соответствующую иконку на панели инструментов (рис. 1.4).
 - 3.8.2. Откроется пустое окно редактора SciNotes (рис. 1.5).

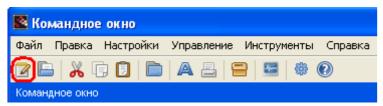


Рис 1.4 – Запуск текстового редактора SciNotes

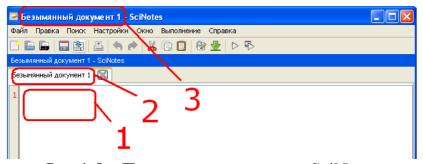


Рис 1.5 – Пустое окно редактора SciNotes

В этом окне (1) можно вводить команды и редактировать их. Команды, введенные здесь, не выполняются сразу, хотя их синтаксис и результаты работы не меняются по сравнению с командной строкой.

Сверху отображается имя текущего файла (2), файл не был сохранен. Вкладки (3) позволяют переключаться между несколькими редактируемыми одновременно файлами.

3.8.3. Введите следующие строки в файл сценария. Они выполняют те же операции, что и в пунктах 3.7.10. и 3.7.11 и предназначены для вывода общей суммы элементов матрицы М и суммы элементов в каждом ее столбце.

```
M=[3 8 23; 5 9 56; 5 9 7];
sum(M)
sum(M,'r')<sup>35</sup>
```

³³ Родное расширение пакета MATLAB *.m. М-файлы, как правило, не могут выполняться в пакете Scilab без предварительного их редактирования и адаптации

³⁴ В версиях пакета Scilab до 5.2 текстовый редактор носил название Scipad и имел другой интерфейс.

³⁵ Не копируйте эти строки в редактор, наберите их самостоятельно, так как могут скопироваться непечатаемые символы.

Обратите внимание, что рядом с именем файла на вкладке появился символ $*^{36}$, означающий, что файл был изменен с момента последнего сохранения (рис. 1.6).

3.8.4. Перед выполнением файла сценария его необходимо сохранить. Выберите из главного меню текстового редактора пункт Файл > Сохранить или соответствующую иконку на панели инструментов редактора, чтобы сохранить файл. Выберите ваш каталог с допуском к лабораторной работе №1 и введите имя файла, например, exl1_1. Обязательно укажите расширение файла (sce), выбрав его из списка. Расширение файлу, в принципе, можно дать любое. Однако, чтобы в дальнейшем не путаться, рекомендуется давать соответствующие расширения пакета Scilab. Нажмите Сохранить (Save).

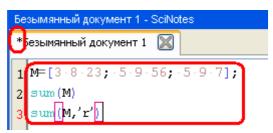


Рис 1.6 – Обозначение изменения файла с момента последнего сохранения

- 3.8.5. В старых версиях сохраненный файл не запустится, пока не будет загружен в среду Scilab³⁷. Чтобы загрузить и выполнить файл сценария, выберите из главного меню текстового редактора пункт Выполнение > Загрузить в Scilab (Execute > Load into Scilab). Результаты работы выводятся в командном окне программы. В некоторых старых версиях программы она может отказаться работать, если путь к файлу содержит символы кириллицы. Обычно в этом случае выводится сообщение, что файл не существует. В этом случае сохраняйте файлы в отдельные каталоги, названные с применением только символов латиницы. Если выводятся другие сообщения об ошибке, то было что-то неверно набрано, проверьте и отредактируйте файл. В новых версиях выберите Выполнение > ...файла с отображением команд для запуска командного файла.
- 3.8.6. Выполнить файл сценария можно и другим способом, используя команду **ехес(имя_файла)** из командной строки. В этом случае необходимо указать полный путь к файлу, что неудобно.
- 3.8.7. Закройте созданный файл сценария. Созданный файл сценария необходимо предъявить преподавателю для получения допуска к выполнению лабораторной работы.

_

³⁶ В предыдущих версиях пакета Scilab появляется слово modified.

³⁷ В пакете MATLAB файлы из рабочего каталога запускаются без какой-либо их загрузки. В пакете Scilab понятия «рабочий каталог» не существует. Файл может быть загружен в среду из любого каталога.

3.9 Комментарии к программам

- 3.9.1. Комментарии в пакете Scilab следуют за знаком //³⁸. Стоящие за этим знаком до конца строки символы не влияют на результат работы программы. Комментарии можно вводить в командной строке, файлах сценария и функциях.
- 3.9.2. Сравните результаты выполнения в командном окне следующих двух команд. Видно, что комментарии не повлияли на результаты расчетов.

```
\rightarrow2+5 ans = 7. 

\rightarrow2+5 //Сложение ans = 7.
```

3.9.3. Создайте новый файл сценария и введите в него следующие команды для расчета корней квадратного уравнения $2x^2 - 13x - 24 = 0$.

3.9.4. Сохраните файл под именем exl1_2.sce и запустите его. Убедитесь, что в окне выводятся правильные результаты. Комментарии были проигнорированы³⁹.

```
First root

x1 = -1.5

Second root

x2 = 8.
```

3.9.5. Закройте созданный файл сценария. Созданный файл сценария необходимо предъявить преподавателю для получения допуска к выполнению лабораторной работы.

4 Требования для получения допуска

Для получения допуска к выполнению лабораторной работы необходимо предъявить преподавателю файлы со следующим содержимым, полученным в

_

³⁸ В пакете MATLAB комментарии следуют за знаком %.

³⁹ В некоторых версиях пакета Scilab все выполняемые строки файла сценария предварительно выводятся на экран. Это зависит от настроек команды **exec**.

результате выполнения примера в пункте 3 (там же описаны требования к отдельным файлам):

- 1. Графический файл с образом экрана компьютера, полученный в результате выполнения действий, описанных в пункте 3.7.
- 2. Файл сценария, полученный в пункте 3.8 (exl1_1.sce), рассчитывающий сумму элементов матрицы.
- 3. Файл сценария, полученный в пункте 3.9 (exl1_2.sce), рассчитывающий корни квадратного уравнения.

Все файлы должны быть помещены в отдельный каталог или архив.

5 Контрольное задание к лабораторной работе №1

5.1 Описание задания

Необходимо составить файл сценария для математического пакета Scilab, который выполняет следующие действия в приведенном ниже порядке:

- создает матрицу A размером NxM, заполненную случайными числами в интервале от k1 до k2;
 - выводит матрицу А на экран;
- вычисляет значение функции f(x) для каждого элемента матрицы A и помещает их в матрицу B;
 - выводит матрицу В на экран.

Значения параметров N, M, k1, k2 и f(x) приведены в таблице 4 в зависимости от варианта.

Для генерации случайных чисел необходимо использовать функцию rand.

Все строки файла должны быть снабжены комментариями.

5.2 Форма отчета

В качестве отчета по лабораторной работе предоставляется в электронном виде файл сценария для математического пакета Scilab, выполняющий указанные в п. 5.1 действия. Файл должен корректно работать и удовлетворять следующим условиям:

- Файл должен содержать в качестве первого комментария комментарий,
 в котором указаны фамилии студентов, номер группы и номер варианта работы.
- Файл должен содержать комментарии ко всем выполняемым действиям.

При защите лабораторной работы могут быть заданы вопросы по любой строке файла сценария и его работе. При защите лабораторной работы должна быть продемонстрирована работоспособность файла сценария.

5.3 Критерии оценки

Следующие погрешности в выполнении работы приводят к снижению оценки за нее:

- отсутствие комментариев в файле сценария или их несоответствие тексту файла сценария;
- использование операторов цикла и ветвления в файле сценария (неоптимальный способ выполнения работы);
- использование функции, отличной от функции rand, для генерации случайных чисел.

5.5 Варианты задания

Табл. 4 Задание по вариантам

№ варианта	N	M	k1	k2	f(x)
1	5	4	$\frac{\pi}{2}$	π	$x^{-2} \cdot \sin(x) \cdot \cos(x) + 5 \cdot \ln(x)$
2	5	3	2,5	3	$\frac{x^2 + 5x + \sqrt{x}}{\sqrt[3]{x}} \cdot e^x$
3	5	2	-1,97	-1,87	$\frac{1+\sin^2(8+x^3)}{\sqrt[3]{8+x^3}}$
4	5	5	5	10	$\left(1 + \lg^2 \frac{x}{4}\right) \left(10 - e^{\frac{x}{4}}\right)$
5	4	4	5	20	$\sqrt[4]{ x^2-2.5 } + \sqrt[3]{\lg(x^2)}$
6	4	3	0,25	0,5	$\frac{10^{x} - 2^{x}}{\lg 8} \sqrt[3]{10}$
7	3	4	0,1	0,5	$(56 + \sin^2 5x)(\arccos(x^2) + e^{-x/2})^{-1}$
8	4	2	3	4	$\frac{\lg(x^2 - 1)}{\log_5(8x^2 - 16)}$
9	2	4	0,1	0,15	$\frac{\arccos\left(x^2 - 0.5^2\right)}{\arcsin\left(x^2 - 0.2^2\right)}$
10	3	3	2,5	3	$3^{x^2-1} - \lg(x^2-1) + \sqrt[3]{x^2-1}$
11	3	2	5	6	$\frac{2\sqrt{x} - 10\log_5 x}{\lg x - 1 }$
12	2	3	11	20	$\sqrt{\frac{ 0.75 - 2x }{\lg^3 x}}$

№ варианта	N	M	k1	k2	f(x)
13	4	5	0,1	0,9	$\left(\arcsin^2 x + \arccos^4 x\right)^3$
14	3	5	$\frac{\pi}{2}$	$\frac{3\pi}{2}$	$\left(\sin^3 x + \cos^3 x\right) \ln x$
15	2	5	10	15	$(2+x)^{-2} \lg^2(2+x)$
16	3	4	10	20	$\frac{\sin x + \cos x}{e^{x + \frac{3}{x}}} \lg(x^2)$
17	2	4	21	23	$x \lg x + \frac{x}{\cos x - \frac{3}{x}}$
18	2	3	$\frac{2\pi}{3}$	π	$\frac{\sin^2(x-\pi)\sqrt{\cos(x+\pi)}}{x^2+5}$
19	6	2	0,1	0,3	$(\arcsin(2x) + \sin x)e^{2x} + 3$
20	2	6	5	6	$(\lg x + \ln x + e^x)(2\sin x + 1)$
21	3	6	0,1	0,5	$5\ln\left(x^2 + 2x - \arcsin(x)\right)$
22	4	6	5	10	$\frac{\sqrt[3]{2x + \log_3(x)}}{2x}$
23	5	6	$\frac{\pi}{3}$	π	$\frac{\sin(x)}{\sin(x+5)+10} + x\lg(x)$
24	6	3	2,1	2,9	$\frac{\arcsin(x-2)}{\sqrt[3]{6x - \arccos(x-2)}}$
25	6	4	5	10	$\frac{2x^5 - 10\log_3 x}{\lg(x+1)}$

ЛАБОРАТОРНАЯ РАБОТА №2. ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ В МАТЕМАТИЧЕСКОМ ПАКЕТЕ SCILAB

1 Основные сведения о работе

1.1 Цель работы

Целью работы является ознакомление с графическими возможностями математического пакета Scilab, предназначенными для визуализации результатов расчетов. В процессе выполнения работы необходимо получить навыки построения двумерных и трехмерных графиков, а также получить представление о дескрипторной графике.

1.2 Порядок выполнения работы

Работа состоит из двух частей. В первой части работы необходимо ознакомиться с теоретическим материалом, включающим общие сведения об особенностях дескрипторной графики, и выполнить практические примеры.

Для получения допуска ко второй части работы необходимо предъявить 5 файлов, полученных в результате выполнения практических примеров, и ответить на теоретические вопросы.

Во второй части работы необходимо выполнить контрольное задание.

1.3 Примечания

Студенты, также работающие в математическом пакете MATLAB, должны обратить внимание на то, что хотя пакет Scilab и является аналогом пакета MATLAB, графические возможности этих двух пакетов существенно различаются. Большинство команд построения графиков пакета Scilab неприменимы в пакете MATLAB, хотя некоторые команды пакета MATLAB в простейшем случае работают в пакете Scilab. Для полноценного использования графических возможностей пакета Scilab не рекомендуется использовать в нем функции пакета MATLAB. Они не рассматриваются в данных методических указаниях.

Несмотря на существенные различия в командах и их синтаксисе, а также в свойствах графических объектов, основные идеи работы с дескрипторной графикой в пакетах Scilab и MATLAB совпадают. В некоторых случаях в примечаниях указываются существенные отличия пакетов MATLAB и Scilab. Студенты, не интересующиеся пакетом MATLAB, могут пропустить эти примечания.

2 Особенности графической подсистемы пакета Scilab

2.1 Графические возможности пакета Scilab

Большинство результатов расчетов требует дальнейшего их графического представления. Это обусловлено, в основном, тем, что графическое представление воспринимается лучше, чем табличное или текстовое. В силу этого математическое программное обеспечение, как правило, включает средства для визуального представления результатов расчетов. К таким средствам относятся, в частности, инструменты для построения графиков, диаграмм, создания анимации.

Графические средства математического пакета Scilab несколько уступают средствам пакета MATLAB⁴⁰, но остаются приемлемыми для получения практически любого желаемого графика. Это обеспечивается широкими возможностями по изменению атрибутов графика: цветов, параметров линий, надписей, шрифтов и пр. Ниже приводится далеко не полный список графических возможностей, которые имеются в пакете Scilab:

- создание графиков в отдельных окнах или в заданном окне;
- использование нескольких окон или построение нескольких графиков в одном окне;
- перемещение окон по экрану или графика внутри окна, изменение их размеров и положения;
 - использование широкого набора цветов;
 - построение двухмерных и трехмерных графиков;
 - построение круговых и столбчатых диаграмм.

2.2 Графические объекты как основа графики пакета Scilab

Язык программирования в пакете Scilab поздних версий является объектно-ориентированным, что позволяет использовать современные средства программирования. Эта особенность пакета Scilab сказывается и на его графических возможностях. Про нее можно забыть при решении простых задач построения, так как были сохранены и традиционные методы работы с графикой.

Задачи построения простых графиков, как правило, решаются с использованием одной команды. Она автоматически устанавливает свойства создаваемых графических объектов. В более сложных задачах использование традиционных методов может существенно усложнить построение или вообще сделать задачу невыполнимой.

Сущность графических объектов можно пояснить следующим образом⁴¹. Каждый элемент графика или диаграммы воспринимается как **объект**,

_

 $^{^{40}}$ Среди всего математического программного обеспечения графические возможности пакета MATLAB полагаются наиболее широкими.

⁴¹ Далее дается упрощенное описание графических объектов для тех, кто не знаком с дескрипторной графикой.

характеризующийся некоторым набором **атрибутов** (параметров, свойств). Такими объектами в пакете Scilab являются, например:

- ломаные линии на графике (polyline),
- поверхность на трехмерном графике (surface),
- текст надписей на графике (text),
- оси координат (axis),
- само графическое окно с рисунком (figure)...

Примером атрибутов объектов могут служить следующие их параметры: цвет линий графика, шрифт текста надписей, толщина линий осей координат, координаты графического окна на экране и пр.

В системах, не являющихся объектно-ориентированными, все атрибуты объектов на графике должны быть указаны до его построения. Дальнейшее их изменение, как правило, представляет некоторые трудности и требует применения отдельной команды для каждого атрибута. В пакете Scilab можно изменять практически любые атрибуты графических объектов уже после построения графика. Часто можно изменить и тип графика.

Большинство объектов зависят друг от друга. Например, при удалении графического окна автоматически удалится и все его содержимое — оси, график и текст в этом окне. При перемещении осей координат внутри окна автоматически переместится и связанный с этими осями график. И, наоборот, оси координат не удастся переместить за пределы графического окна.

Таким образом, существуют объекты более высокого порядка, от которых зависят объекты более низкого порядка. В реальных программах структура объектов может быть очень сложной, но структура графических объектов в пакете Scilab является относительно простой, она представлена на Рис. 2.1.

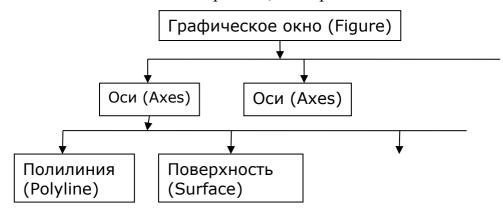


Рис. 2.1 – Упрощенная структура графических объектов в пакете Scilab

Объекты более высокого порядка находятся сверху, более низкого – снизу. Объект более высокого порядка для данного графического объекта в пакете Scilab обозначается как parent, а более низкого – как children. В пакете Scilab каждый графический объект имеет не более одного объекта более высокого порядка и от нуля до разумного числа объектов более низкого порядка.

2.3 Дескрипторная графика

В пакете Scilab существует несколько способов изменить атрибуты (параметры, свойства) графического объекта. Можно:

- Сразу создать графический объект с нужными свойствами, указав их в специальной строке команды создания этого объекта⁴²;
- Изменить отдельный атрибут графического объекта, используя специальную команду для каждого типа атрибута;
- Изменить атрибуты уже построенного графического объекта вручную, используя редактор свойств;
- Изменить атрибуты уже построенного графического объекта напрямую, указав тип атрибута и его значение. Используется операция присваивания.

В простых задачах применимы все указанные способы. В программах (файлах сценария) нельзя менять атрибуты вручную, используя редактор свойств. Редактор свойств применим только при работе в интерактивном режиме. Некоторые атрибуты можно изменить только при прямом обращении к ним.

Рассмотрим, как происходит прямое изменение атрибутов графических объектов. В пакете Scilab каждому графическому объекту ставится в соответствие дескриптор. Дескриптор указывает на графический объект. Он может быть сохранен в переменной, заданной пользователем. Дескриптор используется для указания среде объекта, атрибуты которого будут меняться.

Для смены какого-то атрибута графического объекта в пакете Scilab обычно необходимо указать:

- дескриптор, который на этот объект указывает,
- атрибут, который необходимо изменить,
- новое значение изменяемого атрибута.

Для смены атрибутов используется команда set. Команда get выдает значение заданного атрибута для заданного объекта. Атрибуты в пакете Scilab можно менять и в формате

имя_дескриптора.имя_атрибута=новое_значение_атрибута Графика в пакетах MATLAB и Scilab, использующая дескрипторы, называется дескрипторной графикой (handle graphics).

2.4 Алгоритм построения графиков функций

На плоском мониторе компьютера обычно строят графики функций одной и двух переменных.

Графики функций одной переменной представляются линией или серией линий на координатной плоскости (в декартовых или полярных координатах). Характер линии определяется видом функции.

 $^{^{42}}$ В пакете Scilab, в отличие от пакета MATLAB, таким образом можно установить не все атрибуты

Функции двух переменных представляются в виде поверхности, линии или ряда поверхностей в трехмерном пространстве. Координаты x и y поверхности обычно соответствуют значениям аргументов функции, а z – значению функции. Возможно также представление графика функции двух переменных в виде контурного графика. Он представляет собой проекцию поверхности на плоскость XY с выделением различными цветами различных значений по оси z.

Алгоритмы построения графиков функций одной или двух переменных принципиально не отличаются друг от друга. А именно, график на компьютере строится для конечного числа значений аргументов функции, поэтому необходимо задать эти значения аргумента. Как правило, удобно пользоваться оператором двоеточие:, задавая лишь начальное и конечное значение аргумента, а также шаг. Далее необходимо вычислить значения функции для заданных значений аргументов. После этого полученные массивы значений координат используются для построения графиков.

Для построения графика функции одной переменной y = f(x) необходимо:

- Создать вектор X, состоящий из значений аргументов функции, для которых надо построить график;
 - Вычислить значения у функции в каждой точке вектора аргументов х;
 - Построить график, используя вектора х и у.

Для построения графика функции двух переменных z = f(x, y) необходимо:

- Создать два вектора X и У, каждый из которых содержит значения одной переменной функции;
- Создать две матрицы X и Y одинаковой размерности. Каждый столбец матрицы X представляет собой вектор x. Каждая строка матрицы Y состоит из вектора y. Это обычно делается при помощи специальных функций, но может быть сделано и вручную;
- Вычислить значения функции z для каждой пары аргументов (x,y). Для этого и создавались матрицы X и Y. Они позволяют сочетать каждое значение первой переменной с каждым значением второй переменной, если наложить матрицы друг на друга. Тогда матрица z вычисляется простейшим образом z=f(x,y), где все вычисления внутри функции f(x,y) являются поэлементными. Значения функции также записываются в матрицу, имеющую те же размеры, что и матрицы x и x
 - Построить график, используя матрицы **x**, **y**, **z**⁴⁴.

-

 $^{^{43}}$ Очевидно, что матрицы X и Y имеют одинаковые размеры.

⁴⁴ В пакете MATLAB для построения трехмерных графиков функций, как правило, используются массивы X, Y, z, имеющие одинаковую размерность.

3 Работа с графикой в пакете Scilab

3.1 Подготовка к работе

Для лучшего понимания материала далее, в основном, будет использоваться работа с командной строкой. Однако не запрещается вводить команды в файл сценария. При вводе команд в командной строке при допущении ошибки, возможно, в каждом пункте будет необходимо выполнить все команды заново.

3.1.1. *3anycmume cpedy Scilab*.

3.2 Построение простейшего графика функции одной переменной

Описание задачи

Необходимо построить график функции $y = \sin(x)$ на отрезке $[-\pi; \pi]$.

3.2.1. Вначале необходимо создать вектор, содержащий значения аргумента функции. Шаг разбиения заданного отрезка $[-\pi;\pi]$ желательно выбрать таким, чтобы график на экране выглядел гладким⁴⁵. Слишком мелкий шаг тоже нежелателен. В этом случае будет увеличиваться время, требующееся на построение, а качество графика существенно не изменится. Здесь взят шаг $\frac{\pi}{50}$.

3.2.2. Далее необходимо вычислить значения функции, соответствующие значениям аргумента, и поместить их в другой вектор равной длины. Поместим значения, например, в вектор у.

$$\rightarrow$$
 y=sin(x);

3.2.3. Для построения двумерных графиков используется функция plot2d. В простейшем случае в качестве ее аргументов указываются два вектора: значения аргумента и соответствующие им значения функции. Оба вектора должны иметь равную длину.

$$\rightarrow$$
plot2d(x,y)

График появится в отдельном окне 46 , оно показано на рис 2.2. Не закрывайте его.

3.2.4. Построенный график можно вставить в другое приложение. Чтобы скопировать график в буфер обмена для вставки в другое приложение, например, в редактор Word, выберите из главного меню графика пункт Файл > Копировать в буфер обмена.

⁴⁶ Вид графического окна может различаться в зависимости от версии.

-

 $^{^{45}}$ Соседние точки на графике соединяются прямой линией, поэтому, если точек мало, то график визуально не будет гладким.

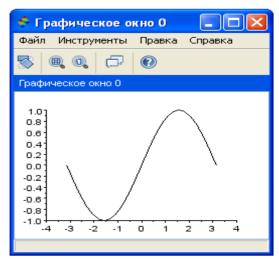


Рис. 2.2 – Построение двумерного графика

- 3.2.5. Вставьте график из буфера обмена в приложение Word или Paint⁴⁷. Сохраните файл с графиком на диске и предъявите его преподавателю для получения допуска к выполнению лабораторной работы.
 - 3.2.6. Закройте графическое окно.
- 3.2.7. Если не указать значения аргумента функции, график которой строится, то в качестве аргумента будет использован номер элемента в векторе. Действительно,

 \rightarrow plot2d(y)

3.2.8. Закройте графическое окно.

3.3 Построение простейшего графика функции двух переменных

Описание задачи

Необходимо построить график функции $z=(x-5)(y+2)-x^2$ для следующего диапазона изменения аргументов: $\begin{cases} 0 \le x \le 2 \\ 1 \le y \le 4 \end{cases}$

3.3.1. Вначале необходимо создать две вектор-строки, содержащие значения аргументов функции. Одна строка — для аргумента x, другая — для аргумента y. Строки могут иметь разную длину. Пусть шаг по оси x составляет 0,1, а по оси y 0,25.

 \rightarrow x=0:0.1:2;

→y=1:0.25:4;

3.3.2. Далее необходимо рассчитать значения функции для каждой парычисел (x, y).

Значения функции записываются в матрицу, число строк которой равно числу столбцов вектора **X**, а число столбцов – числу столбцов вектора **y**.

$$\rightarrow$$
z=(x'-5)*(y+2)-(x.*x)'*ones(y);

 $^{^{47}}$ Рекомендуется использовать форматы *.gif, *.png или любые другие форматы, осуществляющие сжатие без потери информации. Формат *.jpeg не рекомендуется для сохранения графиков.

Здесь используется матричное умножение. А именно $\mathbf{x'}$ дает транспонированную матрицу, превращая вектор-строку в вектор-столбец. Умножение на единичную матрицу, имеющую тот же размер, что и \mathbf{y} , используется, чтобы получить матрицу, каждый столбец которой является копией вектора, содержащего значения \mathbf{x}^{248} .

3.3.3. Для построения графика необходимы ранее полученные матрицы **x**, **y** и **z**. Для построения трехмерных графиков используется команда **plot3d**. График поверхности строится в отдельном окне.

$$\rightarrow$$
plot3d(x,y,z)

3.3.4. Просмотрите график (рис. 2.3) и закройте его.

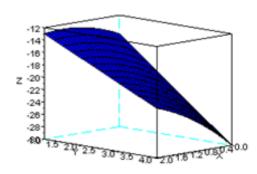


Рис 2.3 – Построение трехмерного графика

3.3.5. Не всегда легко понять, как сконструировать выражение для функции z, чтобы получить нужную матрицу посредством операций над двумя вектор-строками x и y.

Альтернативным способом получения матрицы **Z** является задание двух массивов **X** и **Y**, каждый из которых имеет ту же размерность, что и массив **Z**. Каждый столбец массива **X** состоит из копии транспонированного вектора **x**. Каждая строка массива **Y** состоит из копии вектора **y**. В этом случае массив **Z** получается простыми поэлементными операциями над этими матрицами в соответствии с формулой, задающей функцию. Функция **ndgrid** автоматически создает нужные массивы, хотя их, конечно же, можно создать и вручную.

$$\rightarrow$$
 [X,Y]=ndgrid(x,y)

Просмотрите структуру массивов X и Y.

3.3.6. Далее вычисляются значения массива Z простыми поэлементными операциями

$$\rightarrow$$
Z=(X-5).*(Y+2)-X.^2;

3.3.7. Проверим, что массивы z и Z идентичны. При их вычитании должна получиться нулевая матрица.

⁴⁸ Авторы пакета Scilab полагают, что это простой и понятный способ построения трехмерных графиков, в противоположность способу, используемому в пакете MATLAB. Однако, это утверждение весьма спорно. Определенно, этот способ позволяет экономить оперативную память компьютера. Но способ требует значительных размышлений для получения матрицы. Далее будет дан более простой способ получения матрицы **z**.

 \rightarrow z-Z

3.3.8. Постройте график с новым массивом. Он идентичен графику, построенному в п. 3.3.3. Обратите внимание, что в качестве аргумента функции используются не вспомогательные массивы X и Y, а исходные вектор-строки X и y. Закройте график.

$$\rightarrow$$
plot3d(x,y,Z)

3.4 Использование редактора свойств

Построенный в предыдущем пункте график неудобен для просмотра. На нем отсутствуют подписи осей, а вся поверхность залита синим цветом. Это неудобно при печати на черно-белом принтере, синий цвет будет сливаться с черным цветом сетки.

Отредактировать отдельный график можно, используя редактор свойств графических объектов Graphic Editor (далее он везде называется редактор свойств). В этом случае можно менять свойства графических объектов через графический интерфейс.

3.4.1. Постройте график функции $y = \ln(x)$ на отрезке [0,5; 2] с шагом разбиения по оси х 0,05 (рис. 2.4)

```
\rightarrowx=0.5:0.05:2;

\rightarrowy=log(x);

\rightarrowplot2d(x,y)
```

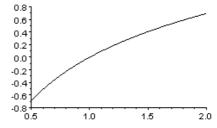


Рис. $2.4 - \Gamma$ рафик функции $y = \ln(x)$

- 3.4.2. Выберите в меню построенного графического окна пункт Правка > Свойства графического окна... (Figure properties)
 - 3.4.3. Появится редактор свойств графических объектов (рис 2.5).
- 3.4.4. Сейчас в разделе Objects Browser (1) выделен графический объект Figure, т.е. само окно с графиком. Чтобы изменить его атрибуты, достаточно задать новые их значения в разделе Object Properties (2), и изменения сразу вступят в силу.

Измените заголовок окна, набрав на вкладке Style в строке Figure name слова Мой график %d или My graph %d (рис. 2.6).

Символ %d служит для вывода номера окна. По умолчанию первому окну присваивается номер 0.

Измените высоту окна, набрав 560 в строке Y axis size. Измените цвет фона окна, установив цвет номер 12 движком Back. Color. Соответствие цветов и их номеров можно изменить на вкладке Colormap (палитра).

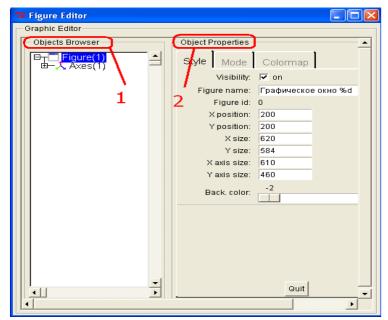


Рис. 2.5 – Редактор свойств графических объектов

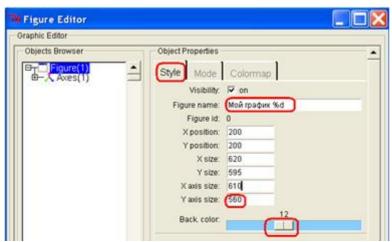


Рис. 2.6 – Изменение заголовка окна

- 3.4.5. Переключитесь на графическое окно. Убедитесь, что оно изменилось. Результат приведен на рис. 2.7^{49} .
- 3.4.6. В разделе Objects Browser выберите объект Axes, чтобы поменять свойства этого объекта осей координат. Обратите внимание, что окно Figure Editor изменилось на Axes Editor. Чтобы ввести название оси x, введите "x" в поле Text. Увеличьте размер шрифта до четырех в Font size/color. Выберите черный цвет сетки по оси x в Grid color. Если этот цвет принимает отрицательное значение, то сетка не будет отображаться, как было до этого.

Последовательность действий проиллюстрирована на рисунке 2.8.

42

⁴⁹ Размеры окна могут немного отличаться в зависимости от настроек программы и монитора.

3.4.7. Просмотрите график еще раз (рис. 2.9), отметив изменения на нем.

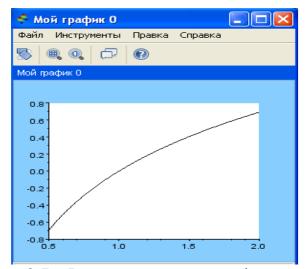


Рис. 2.7 – Результат изменения фона окна

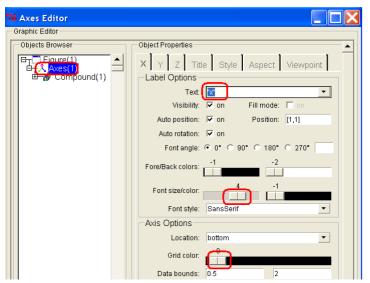


Рис 2.8 – Последовательность изменений свойств графика

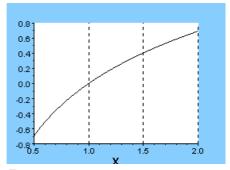


Рис. 2.9 – Видоизмененный двумерный график

3.4.8. Переключитесь на вкладку Y (рис 2.10).



Рис. 2.10 – Вкладка Ү

3.4.9. Установите аналогичные параметры для оси Y так, чтобы получить рисунок 2.11, приведенный ниже.

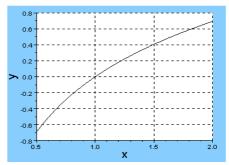


Рис. 2.11 – Видоизмененный график для оси Ү

3.4.10. Переключитесь на вкладку Title (рис 2.12). Здесь устанавливаются параметры заголовка графика. Введите сам заголовок "y=ln(x)" в поле Text. Установите высоту символов 5.

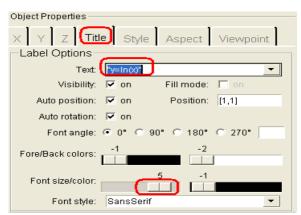


Рис. 2.12 – Вкладка Title

3.4.11. Переключитесь в разделе Object Browser на объект Compound (рис. 2.13).



Рис. 2.13 – Объект Compound

Объект Compound объединяет однотипные объекты, для которых объектом более высокого порядка является объект ахеs. Это важно, так как на

графике может быть не одна линия, а две или больше. В данном случае линия одна. Переключитесь на объект Polyline.

3.4.12. Измените стиль линии Line на штриховую dash, цвет линии на красный Foreground (5), стиль маркера на графике Mark style на крест cross, а цвет маркера Mark foreground на цвет 20, как на рисунке 2.14.

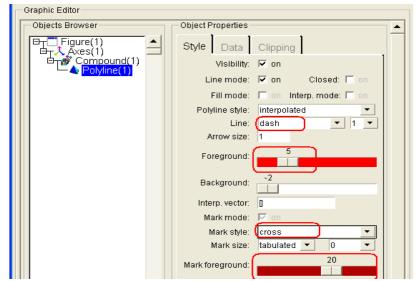


Рис. 2.14 – Изменение стиля и цвета линий и маркеров

- 3.4.13. Результирующий график приведен на рисунке 2.15.
- 3.4.14. Для того, чтобы напрямую сохранить график в файл, не используя копирование в буфер обмена, выберите из меню графика пункт Файл > Экспортировать...

Выберите из предложенного списка графических форматов подходящий (bmp, gif или jpeg)⁵⁰ и сохраните файл под именем, например, exl2_2. Сохраненный файл необходимо предъявить преподавателю для получения допуска к выполнению лабораторной работы.

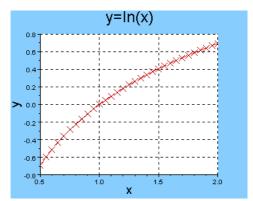


Рис. 2.15 – Результирующий график

45

 $^{^{50}}$ Рекомендуется использовать формат *.gif. Файлы этого формата занимают наименьшее место без потери качества изображения (теряются только некоторые данные о цвете).

- 3.4.15. При желании измените другие атрибуты графических объектов и посмотрите на результат.
- 3.4.16. Закройте графическое окно и редактор свойств. Это действие является необратимым, все несохраненные данные и настройки графического окна будут потеряны.

3.5 Дескрипторная графика

Использование редактора свойств оправдано, только если необходимо построить один график. Если же необходимо строить множество графиков со схожими параметрами, то удобнее использовать средства, позволяющие менять атрибуты графика средствами программирования. Так как в противном случае, при использовании редактора свойств одни и те же параметры графика придется устанавливать каждый раз заново.

Основной проблемой при изменении атрибутов графических объектов средствами программирования является получение дескриптора этих объектов⁵¹. Дескриптор нужен для указания на объект.

3.5.1. Создать графическое окно с одновременным получением его дескриптора можно командой SCf⁵². В качестве аргумента функции можно указать идентификатор окна (целое положительное число). По умолчанию идентификатор присваивается открытым окнам по порядку, начиная с нуля. Функция возвращает дескриптор этого окна. Создайте окно с номером 2 и поместите его дескриптор в переменную fig2.

```
\rightarrowfig2=scf(2);
```

Обратите внимание, что новое окно имеет номер 2 (рис. 2.16). Новое окно становится текущим. Не закрывайте его.

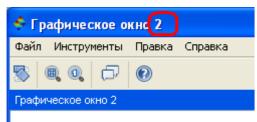


Рис. 2.16 – Текущее окно

3.5.2. Чтобы просмотреть атрибуты существующего окна и их значения, просто выведите значение переменной fig2.

→fig2 fig2 = Handle of type "Figure" with properties:

⁵¹ В пакете MATLAB получить дескриптор объекта гораздо проще, так как при создании любого объекта можно сразу поместить его дескриптор в переменную. В пакете Scilab это возможно не всегда.

⁵² В пакете MATLAB подобные действия выполняет команда figure

```
children: "Axes"
figure position = [200, 200]
figure size = [620,584]
axes size = [610, 460]
auto_resize = "on"
viewport = [0,0]
figure name = "Графическое окно %d"
figure id = 2
info message = ""
color map= matrix 32x3
pixmap = "off"
pixel drawing mode = "copy"
anti aliasing = "off"
immediate drawing = "on"
background = -2
visible = "on"
rotation style = "unary"
event handler = ""
event handler enable = "off"
user data = []
tag = «»
```

Возможные свойства графического окна и их значения можно найти в помощи в разделе Graphics > figure_operations > figure_properties. В различных версиях этот раздел помощи может располагаться в разных местах, поэтому можно просто ввести в командную строку команду help figure_properties. Некоторые свойства уже рассматривались в пп. 3.4.4 – 3.4.5.

3.5.3. Любое свойство можно изменить без особых затруднений. Например, для изменения размера графического окна (обратите внимание на сноски, обозначенные надстрочными индексами) наберите

```
\rightarrow fig2<sup>53</sup>.figure_size<sup>54</sup>=[500,400]<sup>55</sup>;
```

Окно сразу же уменьшилось.

3.5.4. Для создания новых координатных осей с одновременным помещением их дескриптора в переменную а наберите следующее

```
→a=newaxes()<sup>56</sup>
a =
Handle of type "Axes" with properties:
```

```
parent: Figure
children: []
visible = "on"
axes_visible = ["off","off","off"]
```

⁵³ Дескриптор объекта, в данном случае графического окна.

⁵⁴ Обозначение атрибута, можно посмотреть в п. 3.5.2.

⁵⁵ Новое значение атрибута.

⁵⁶ Здесь умышленно не поставлена точка с запятой, чтобы сразу вывести атрибуты осей.

```
axes_reverse = ["off","off","off"]
grid = [-1, -1]
grid position = "background"
x location = "bottom"
y location = "left"
title: "Label"
x label: "Label"
y label: "Label"
z label: "Label"
auto ticks = ["on","on","on"]
x ticks.locations = matrix 11x1
y ticks.locations = matrix 11x1
z ticks.locations = []
x ticks.labels = matrix 11x1
y ticks.labels = matrix 11x1
z ticks.labels = []
box = "off"
filled = "on"
sub ticks = [1,1]
font style = 6
font size = 1
font color = -1
fractional font = "off"
isoview = "off"
cube scaling = "off"
view = "2d"
rotation angles = [0,270]
log flags = "nnn"
tight limits = "off"
data bounds = [0,0;1,1]
zoom\ box = []
margins = [0.125, 0.125, 0.125, 0.125]
axes bounds = [0,0,1,1]
auto clear = "off"
auto scale = "on"
hidden axis color = 4
hiddencolor = 4
line mode = "on"
line style = 0
thickness = 1
mark mode = "off"
mark style = 0
mark size unit = "tabulated"
mark size = 0
```

```
mark foreground = -1
mark background = -2
foreground = -1
background = -2
arc drawing method = "lines"
clip state = "off"
clip box = []
user data = []
```

- 3.5.5. Значения атрибутов осей можно просмотреть в помощи
- →help axes properties
- 3.5.6. Новые оси стали текущими. Новый график построится в этих осях. Постройте график функции $y = \cos(x)$ на отрезке $[0; 2\pi]$.

```
→x=0 :%pi/50 :2*%pi;
\rightarrowy=cos(x);
\rightarrowplot2d(x,y)
```

3.5.7. Для изменения свойств линии графика необходимо получить ее дескриптор. Это можно сделать следующим образом. Чтобы получить дескриптор объекта более низкого порядка, чем оси, необходимо набрать

```
\rightarrowc=a.children
```

Очевидно, что это объект – compound, объединяющий несколько линий графика (см. п. 3.4.11). В данном случае линия только одна. Поэтому ее дескриптор получается аналогичным образом.

```
→pl=c.children<sup>57</sup>
Handle of type "Polyline" with properties:
```

```
parent: Compound
children: []
visible = "on"
data = matrix 101x2
closed = "off"
line mode = "on"
fill mode = "off"
line style^{58} = 0
thickness^{59} = 1
arrow size factor = 1
polyline style^{60} = 1
foreground^{61} = 1
background = -2
interp color vector = []
```

⁵⁷ Если линий несколько, то дескриптор, например, линии 2 получается как pl2=c.children(2)

⁵⁸ Стиль линии – сплошная, штриховая и пр.

⁵⁹ Толщина линии.

⁶⁰ Тип самого графика.

⁶¹ Цвет линии, соответствие номера и цвета определяется палитрой графического окна.

```
interp_color_mode = "off"
mark_mode<sup>62</sup> = "off"
mark_style<sup>63</sup> = 0
mark_size_unit = "tabulated"
mark_size = 0
mark_foreground<sup>64</sup> = -1
mark_background = -2
x_shift = []
y_shift = []
z_shift = []
bar_width = 0
clip_state = "clipgrf"
clip_box = []
user_data = []
```

3.5.8. Просмотреть возможные свойства полилинии графика можно, набрав

```
→help polyline properties
```

3.5.9. Просмотреть значение любого атрибута, соответствующего объекту, описываемому дескриптором, можно двумя почти равнозначными способами. Значение можно присвоить переменной. Сравните:

```
→pl.foreground
ans = 1.
→get(pl,"foreground")<sup>65</sup>
ans = 1.
```

Указывает, что цвет линии имеет номер 1 в текущей палитре.

- 3.5.10. Просмотрите значения еще двух-трех самостоятельно выбранных атрибутов.
- 3.5.11. Изменить значение атрибута можно также двумя способами. Один уже был описан ранее, в п. 3.5.3. Измените цвет графика на синий в стандартной палитре. Обозначения некоторых цветов в стандартной палитре приведены в Табл. 5. Однако учтите, что палитра может быть изменена различными командами.

```
\rightarrowpl.foreground=2;
```

Просмотрите полученный график, а теперь измените цвет на красный, другим способом:

```
→set(pl,"foreground",5)
```

Просмотрите полученный график. Не закрывайте его.

⁶² Наличие маркера точек

⁶³ Стиль маркера

⁶⁴ Цвет маркера

⁶⁵ Этот способ совместим с пакетом MATLAB

Табл. 5 Номера цветов в стандартной палитре пакета Scilab

Номер	Цвет	Номер	Цвет
1	черный, black	5	красный, red
2	синий, blue	6	пурпурный, magenta
3	зеленый, green	7	желтый, y ellow
4	голубой, суап	8	белый, white

3.5.12. Изменить тип маркера на графике можно, не включая режим отображения маркера mark_mode. Он включится автоматически. Возможные значения для атрибута mark_style (стиль маркера) приведены в Табл. 6.

Измените маркер на плюс (рис.2.17).

$$\rightarrow$$
pl.mark_style=1;

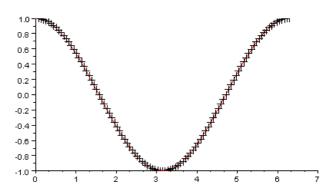


Рис. 2.17 – Результат изменения стиля маркера на «плюс»

Табл. 6 Стили маркеров в пакете Scilab

No	Название маркера	Рисунок	№	Название маркера	Рисунок
0	Точка	•	8	Крестик в ромбе	
1	Плюс	+	9	Кружок	0
2	Крестик	×	10	Звездочка	*
3	Перечеркнутый кружок	0	11	Квадрат	
4	Залитый ромб	•	12	Треугольник острием вправо	\triangle
5	Незалитый ромб	♦	13	Треугольник острием влево	\triangleleft
6	Треугольник острием вверх	Δ	14	Звезда	☆
7	Треугольник острием вниз	∇			

3.5.13. Обратите внимание, что свойство mark_mode автоматически изменилось на "on".

```
\rightarrowpl.mark_mode ans = on
```

- 3.5.14. Теперь поменяйте цвет маркера точек графика на красный.
- \rightarrow pl.mark foreground=5;
- 3.5.15. Свойства объекта можно менять в любой момент времени. Например, можно поменять параметры ранее созданных осей, дескриптор которых был сохранен ранее в переменной а. Измените цвет линий сетки по оси у на черный, а по х на желтый. Тем самым сетка станет видимой.
 - \rightarrow a.grid=[7 1];
- 3.5.16. Смените цвет окантовки окна на голубой. Дескриптор окна был ранее сохранен в переменной fig2.
 - →fig2.background=4;
- 3.5.17. Сохраните полученный график (рис. 2.18) в графическом файле exl2_3 и предъявите его преподавателю для получения допуска к выполнению лабораторной работы.

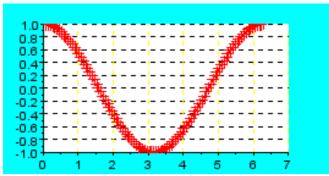


Рис 2.18 – Результирующий график

3.5.18. Закройте окно и удалите ненужные переменные a, fig2 и pl при помощи команды clear. При необходимости обратитесь к лабораторной работе N21.

3.6 Команды для изменения графика и построения нескольких графиков

Для изменения свойств графических объектов не обязательно использовать дескрипторы. Этого можно добиться и другими методами. Они описаны ниже.

3.6.1. При построении графика можно сразу указать требуемый цвет линии:

```
→x=-%pi :%pi/5 :%pi ;
→y=sin(x)+cos(x) ;
→plot2d(x,y,style=3) ;
```

Сразу строится график зеленого цвета (рис. 2.19). Закройте окно с графиком.

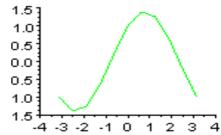


Рис. 2.19 – Результат построения графика зеленого цвета

3.6.2. Если параметр style отрицательный, то его значение указывает не на цвет линии, а на тип маркера.

$$\rightarrow$$
plot2d(x,y,style=-3);

Построился точечный график (рис 2.20). Просмотрите его и закройте.

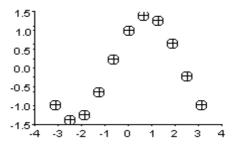


Рис. 2.20 – Точечный график

3.6.3. Чтобы сразу построить две линии в одном окне, необходимо в качестве второго аргумента функции plot2d указать матрицу, каждый столбец которой соответствует значениям соответствующей функции.

Постройте на одних осях и в одном окне графики функций y = x и y = x + 3 на отрезке [1; 10]. В параметре style укажите вектор, в котором содержатся параметры построения для каждого столбца.

```
→x=1 :0.1 :10 ;
→y1=x ;
→y2=x+3 ;
→plot2d(x,[y1' y2'],style=[2 4]) ;
```

Здесь необходимая для второго аргумента функции матрица была сформирована прямо при указании аргумента функции. Также сразу указаны цвета линий графика. Не закрывайте график.

3.6.4. При построении линий графика необходимое окно и ось координат были построены автоматически (рис. 2.21). Чтобы теперь узнать дескриптор построенных объектов, можно воспользоваться функциями gca и gcf.

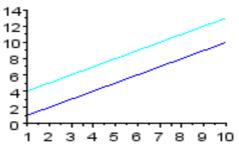


Рис 2.21 – Результат построения двух линий в одном окне

Функция gca возвращает дескриптор текущих осей, gcf — дескриптор текущего графического окна. Функции аргументов не имеют, но при вызове любой функции указываются круглые скобки, даже если у нее нет аргументов.

- →f1=gcf();
 →a1=gca();
- 3.6.5. Дескрипторы линий графика можно получить так же, как это было сделано в п. 3.5.7. Однако можно это сделать одной командой. Обратите внимание, что линий две.
- ightarrow11=a1.children.children(1); //дескриптор первой линии 66
- →12=a1.children.children(2); //дескриптор второй линии
- 3.6.6. Теперь можно изменить стиль первой линии. Различные типы линий задаются атрибутом line_style. Типы линий в пакете Scilab приведены в таблице 7.

Табл. 7. Типы линий в пакете Scilab

Nº	Название типа линии	Рисунок	№	Название типа линии	Рисунок
0,1	Сплошная		5	Штрих – две точки	- · · - · ·
2	Крупный штрих		6	Крупный штрих – мелкий штрих	
3	Мелкий штрих		7	Пунктирная	
4	Штрихпунктирная	- · - ·	8	Двойной пунктир	

- \rightarrow 11.line style=4;
- 3.6.7. Измените толщину второй линии (толщина линии указывается в пикселях).
 - \rightarrow 12.thickness=2;

Просмотрите результат (рис. 2.22). Закройте окно с графиком.

⁶⁶ Комментарии, конечно же, вводить в командную строку не обязательно.

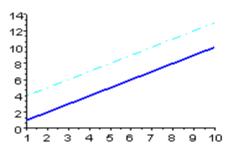


Рис. 2.22 – Результирующий график с видоизмененной первой линией

3.6.8. Недостатком описанного подхода к построению двух графиков в одних осях является невозможность построения графиков, зависящих от разных аргументов. Это можно сделать повторным применением функции plot2d.

```
→x1=0:0.1:1;
→x2=0.5:0.05:1.2;
→y1=2*x1;
→y2=3*(x2-1);
→plot2d(x1, y1, style=5)
→plot2d(x2, y2, style=2)
Два графика построились в одном окне (рис.2.23).
```

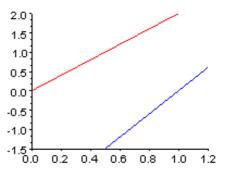


Рис. 2.23 – результат построения двух линий на одних осях

3.6.9. Обратите внимание, что второй график построился в том же окне, что и предыдущий. Закройте это окно. Чтобы построить два графика, каждый в отдельном окне, необходимо вначале создать эти окна и переключаться между ними при необходимости.

Создается первое окно, на которое указывает дескриптор f1.

$$\rightarrow$$
f1=scf(1);

Создается второе окно, на которое указывает дескриптор f2.

$$\rightarrow$$
f2=scf(2);

3.6.10. Переключитесь на окно 1. Для этого используется та же функция **scf.** Постройте график в этом окне.

Переключитесь на окно 2 и постройте график в этом окне.

```
→scf(f2);
→plot2d(x2,y2,style=2)
```

Таким образом, было построено два различных графика в различных окнах (рис.2.24). Закройте все графические окна.

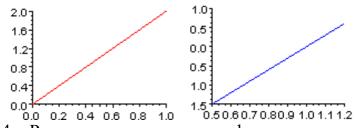


Рис. 2.24 – Результат построения графиков в различных окнах

3.6.11. Часто нужно построить графики в одном окне, но на разных осях координат. Для создания нескольких осей координат в одном окне используют функцию subplot(mnp). Она создает оси координат на позиции с номером p, причем оси в окне расположены в виде матрицы размера $m \times n$. Отсчет позиции ведется по направлению слева направо и сверху вниз.

Постройте четыре графика y1 = f(x1) в одном окне, причем линии на графиках должны быть различных цветов. Проконтролируйте результат выполнения каждой команды.

- \rightarrow f3=scf(3); //Создается новое окно.
- \rightarrow subplot(221) //Создаются оси для левого верхнего графика
- \rightarrow plot2d(x1,y1,style=5) //Строится график в текущих осях
- \rightarrow a1=gca(); //Для дальнейшего использования сохраняется дескриптор текущих осей
- \rightarrow subplot(222) //Создаются оси для правого верхнего графика
- \rightarrow plot2d(x1,y1,style=3) //Новые оси стали текущими, в них и строится новый график.
 - →a2=qca(); //и.т.д.
 - \rightarrow subplot (223)
 - \rightarrow plot2d(x1,y1,style=6)
 - \rightarrow a3=gca();
 - \rightarrow subplot (224)
 - \rightarrow plot2d(x1,y1,style=1)
 - \rightarrow a4=qca();

Результат построений приведен на рисунке 2.25.

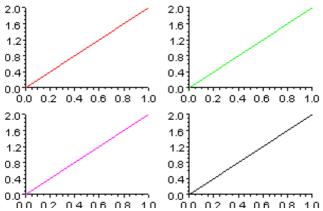


Рис. 2.25 – Результат построения четырех графиков в одном окне

- 3.6.12. Измените цвет фона верхних правых осей на синий. Для этого используется полученный ранее дескриптор а2.
 - \rightarrow a2.background=2;
- 3.6.13. Теперь замените нижний левый график на график y2 = f(x2). Для этого вначале нужно удалить старый график. При удалении графического объекта удаляются и все объекты более низкого порядка.

Далее переключитесь на оси 3 (сделайте их текущими), для этого используется функция sca. Постройте график на этих осях.

$$\rightarrow$$
plot2d(x2,y2,style=2)

Результат построений приведен на рисунке 2.26. Аналогично можно менять и другие атрибуты графиков. Не закрывайте окно с графиками.

3.6.14. На осях 3 можно построить еще один график. Значения функции можно вычислить сразу при задании ее аргумента. Например, постройте график функции $y = \sin(x)$ для существующего вектора аргументов **X1**.

$$\rightarrow$$
plot2d(x1,sin(x1),style=5)

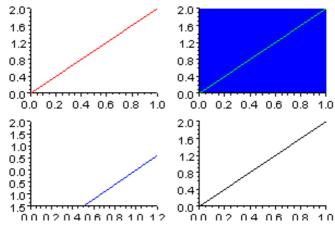


Рис. 2.26 – Результат изменения левого нижнего графика

3.6.15. Для различения графиков, построенных на одних осях, можно использовать легенду. При этом в качестве аргументов указываются надписи для каждой линии графика.

 \rightarrow legend('y1=f(x)','y2=f(x)');

Результат построений приведен на рисунке 2.27.

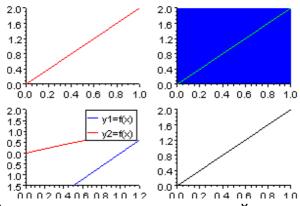


Рис. 2.27 – Результат применения надписей для линий графика

3.6.16. На осях можно вывести заголовок. Можно для этого использовать и изменение атрибутов осей, но обычно проще сделать следующим образом:

→title('My graph3')

3.6.17. Обратите внимание, что дескриптор легенды можно получить из атрибутов осей а3.

→a3

a3 =

Handle of type "Axes" with properties:

parent: Figure
children: ["Legend";"Compound";"Compound"]
...

- 3.6.18. Сохраните полученный график в графическом файле **ex12_4** и предъявите его преподавателю для получения допуска к выполнению лабораторной работы.
- 3.6.19. Закройте окно с построенными графиками. Удалите все ненужные переменные.

→clear

3.7 Различные виды графиков в Scilab

3.7.1. В пакете Scilab возможно создание и других типов графиков и диаграмм (рис.2.27 — 2.32). Далее эти графики рассматриваются лишь поверхностно. Желательно просмотреть помощь по каждой из рассмотренных команд. Не забывайте закрывать или удалять предыдущий график функции перед построением нового. Для этого можно использовать команду clf или делать это вручную.

Для дальнейших построений создайте вектор из двадцати случайных значений и вектор, содержащий числа от 5 до 24.

 \rightarrow t=5:24;

 \rightarrow v1=rand(1,20);

3.7.2. Ступенчатый график функции (рис. 2.27)⁶⁷

 \rightarrow plot2d2(t,v1)

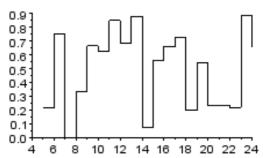


Рис. 2.27 – Ступенчатый график функции

3.7.3. График, построенный в виде вертикальных линий (рис. 2.28)⁶⁸ \rightarrow plot2d3 (t, v1)

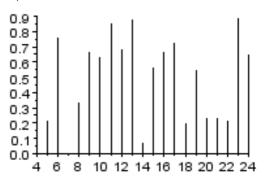


Рис. 2.28 – График, построенный в виде вертикальных линий

3.7.4. Столбчатая диаграмма (рис 2.29).

→bar(v1)

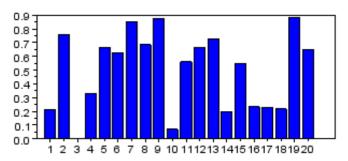


Рис. 2.29 – Столбчатая диаграмма.

 $^{^{67}}$ Здесь и далее конкретный полученный вид графика будет отличаться в зависимости от сгенерированных случайных значений.

⁶⁸ Не забудьте очистить графическое окно от предыдущего графика командой **clf** или иным способом.

3.7.5. Постройте столбчатую диаграмму с заданными значениями по оси (рис. 2.30).

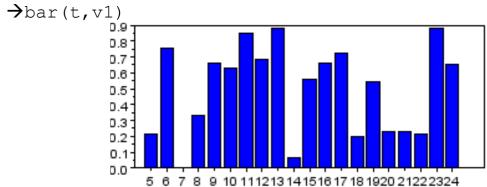


Рис. 2.30 – Столбчатая диаграмма с заданными значениями по оси.

- 3.7.6. Постройте горизонтальную столбчатую диаграмму (рис.2.31).
- \rightarrow barh(v1)
- 3.7.7. Постройте круговую диаграмму (рис.2.32).
- \rightarrow pie(v1)
- 3.7.8. По желанию, просмотрите помощь по приведенным командам и постройте другие графики самостоятельно. Закройте все открытые графические окна.
- 3.7.9. Графики функций иногда удобнее строить с использованием логарифмической шкалы по одной или обеим осям координат. Например, построим график функции $y = e^x$ на участке [0; 20].

```
\rightarrowx=0:0.1:20;

\rightarrowy=exp(x);

\rightarrowplot2d(x,y)
```

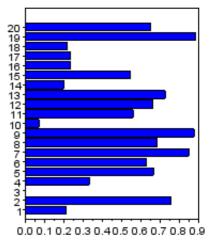


Рис. 2.31 – Горизонтальная столбчатая диаграмма

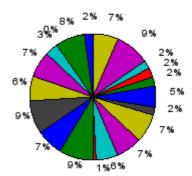


Рис. 2.32 – Круговая диаграмма

3.7.10. Полученный график не очень удобен в использовании из-за резкого роста значения функции, удалите его.

 \rightarrow clf

3.7.11. Для построения графика с логарифмическим масштабом осей (рис. 2.33) используется параметр logflag. Этот параметр равен 'nn' для обычного графика, 'nl' для логарифмического масштаба по оси у, 'ln' для логарифмического масштаба по обеим осям.

3.7.12. Закройте график.

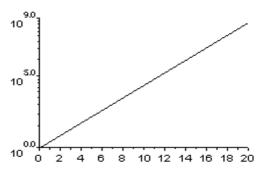


Рис. 2.33 – График с логарифмическим масштабом осей

3.8 Графики функций двух переменных

Атрибуты трехмерных графиков можно менять так же, как и атрибуты двумерных графиков. Постройте график функции $z = x^2 - y^3 + 2xy$ для диапазонов изменения аргументов $-1 \le x \le 1$ и $0 \le y \le 1$.

3.8.1. Задайте вектора, определяющие диапазон изменения аргументов функции:

 \rightarrow x=-1:0.05:1;

 \rightarrow y=0:0.05:1;

3.8.2. Автоматически сгенерируйте вспомогательные вектора для определения значений функции.

 \rightarrow [X,Y]=ndgrid(x,y);

3.8.3. Вычислите матрицу значений функции.

 \rightarrow Z=X.^2-Y.^3+2*X.*Y;

3.8.4. Постройте график залитой цветом поверхности.

 \rightarrow plot3d(x,y,Z)

График приведен на рисунке 2.34.

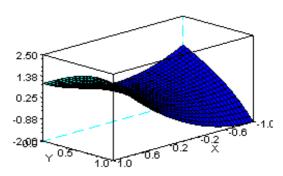


Рис. 2.34 — Результат построения графика с залитой поверхностью

- 3.8.5. Просмотрите атрибуты линий поверхности в Graphic Editor. Их можно также менять, как и в двумерных графиках. По желанию, поменяйте некоторые атрибуты графика. Видно, что возможных атрибутов меньше, чем для двухмерных графиков 69 .
 - 3.8.6. Закройте окно с графиком.
- 3.8.7. Далее будут продемонстрированы различные типы трехмерных графиков на примере графика функции $z = \sqrt{x^2 + y^2}$ для аргументов $0 \le x \le 1$ и $1 \le y \le 2$. Вначале рассчитываются значения функции.

```
\rightarrowx=0 :0.05 :1 ;

\rightarrowy=1 :0.05 :2 ;

\rightarrow[X,Y]=ndgrid(x,y) ;

\rightarrowZ=sqrt(X.^2+Y.^2);
```

3.8.8. Далее создается графическое окно, первые оси координат, и строится каркасный график. Сохраняются их дескрипторы. Обратите внимание на сноски, обозначенные надстрочными индексами.

```
→f1=scf(1);
→subplot(221)
→a1=gca();
→plot3d(x,y,Z,flag=[0<sup>70</sup> 2<sup>71</sup> 4<sup>72</sup>])
```

3.8.9. Создаются вторые оси координат, и строится график поверхности, залитой синим цветом, без подписей по координатным осям.

```
→subplot(222)
→a2=gca();
```

⁶⁹ Студенты часто игнорируют это задание. Однако, понимание того, какие атрибуты есть у трехмерных графиков важно для выполнения индивидуального задания.

⁷⁰ Параметр соответствует каркасному графику.

⁷¹ Автоматическое масштабирование.

⁷² Рисуются координатные оси с надписями по осям.

- \rightarrow plot3d(x,y,Z,flag=[-2⁷³ 2 3⁷⁴])
- 3.8.10. Создаются третьи оси координат, и строится контурный график функции с восемью контурами, выбираемыми автоматически.
 - \rightarrow subplot (223)
 - \rightarrow a3=gca();
 - \rightarrow contour (x, y, Z, 8)

В построенном контурном графике пространство между контурами цветом не залито. График с заливкой цветом строится функцией contour.

- 3.8.11. Создаются четвертые оси координат, и строится график поверхности, залитой цветом в зависимости от значения функции.
 - \rightarrow subplot (224)
 - \rightarrow a4=qca();
 - \rightarrow plot3d1(x,y,Z)

Результат построений приведен на рисунке 2.35.

- 3.8.12. Четвертый график с каркасными линиями выглядит не очень красиво. Уберите эти линии. Для начала сохраните дескриптор этой поверхности, просмотрев возможные ее атрибуты.
 - →sur4=a4.children

Измените стиль изображения графика функции.

→sur4.color mode=-1;

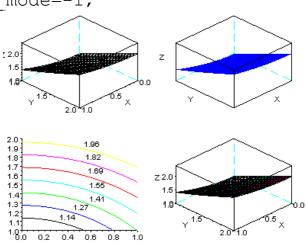


Рис. 2.35 – Результирующие графики

- 3.8.13. В принципе, можно изменить тип и уже существующего графика.
- →sur2=a2.children;
- →sur2.color_flag=1;
- 3.8.14. Измените палитру графиков на черно-белую, удобную для печати на лазерном принтере. Обратите внимание, что палитра это свойство графического окна, она применяется сразу ко всем графикам в окне.

$$\rightarrow$$
f=gcf();

⁷³ Параметр соответствует заливке синим цветом без каркасных линий (положительное значение параметра дает заливку цветом с каркасными линиями).

⁷⁴ Нет подписей по координатным осям.

 \rightarrow f.color map=graycolormap(32);

Здесь функция graycolormap(32) возвращает палитру в серых тонах. Можете посмотреть другие палитры: autumncolormap(32), bonecolormap(32), coolcolormap(32), coppercolormap(32) $u.m.\partial$. \mathcal{I}_{pyzue} палитры можно найти в помощи. Вернитесь к палитре graycolormap(32). Наибольший интерес также представляют палитры jetcolormap(32) и hotcolormap(32), которые дают плавный переход от одного цвета к другому.

3.8.15. Чтобы вывести соответствие цветов и значений функции на графике 2, нужно вначале переключиться на соответствующие оси. Затем используется функция colorbar, аргументами которой являются минимальное и максимальное значение для контуров. Логично использовать минимальное и максимальное значение функции, график которой строится.

```
\rightarrowsca(a2);
\rightarrow colorbar (min(Z), max(Z))
```

3.8.16. Присвойте заголовок графику 2. Можно не переключаться на соответствующие оси, а просто указать их дескриптор в качестве аргумента функции.

```
→title(a2,'График2')
или, для систем, плохо поддерживающих кириллицу,
→title(a2,'Graph2')
```

3.8.17. Присвоить имена осям и заголовок графику можно одной командой xtitle.

```
\rightarrowsca(a1);
→xtitle('График1','ось х','ось y','ось z')
Окончательный результат построений приведен на рисунке 2.36.
```

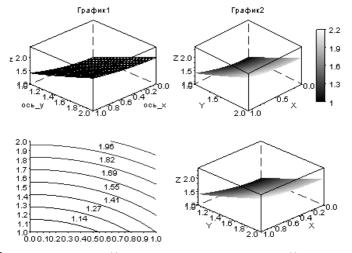


Рис. 2.36 – Окончательный результат построений четырех графиков

4 Требования для получения допуска

Для получения допуска к выполнению лабораторной работы необходимо предъявить преподавателю файлы со следующим содержимым, полученным в результате выполнения примера в пункте 3 (там же описаны требования к отдельным файлам):

- 1. Графический файл или файл формата *.doc, содержащий график, построенный в пункте 3.2.
- 2. Графический файл с результатом работы с редактором свойств в пункте 3.4.
 - 3. Графический файл с результатами работы в пункте 3.5.
- 4. Графический файл с четырьмя двумерными графиками в одном окне, полученный в пункте 3.6.
- 5. Графический файл с четырьмя трехмерными графиками в одном окне, полученный в пункте 3.8.

Все файлы должны быть помещены в отдельный каталог или архив.

5 Контрольное задание к Лабораторной работе № 2

5.1 Описание задания

Необходимо составить файл сценария для математического пакета Scilab, в результате выполнения которого на экран будет выведено графическое окно, содержащие графики трех функций $a = f_1(b)$, $a = f_2(b)$, $z = f_3(x,y)$. Функции в зависимости от варианта приведены в таблицах 8-9 в конце задания, там же приведены диапазоны изменения аргументов функций. Результаты построений необходимо сохранить в файле графического формата.

Результаты выполнения файла сценария и сам файл сценария должны удовлетворять следующим условиям:

- Все графики функций должны располагаться в **одном** окне на **двух** координатных осях. В графическом окне не должно оставаться пустых мест, занимающих больше $\frac{1}{4}$ части графического окна.
- На **верхних** координатных осях должны быть построены графики функций $a = f_1(b)$ и $a = f_2(b)$, на нижних оставшийся график функции $z = f_3(x, y)$.
- Графики различных функций на верхних осях должны иметь различные цвета. Графики должны быть подписаны и иметь легенду.
- Все координатные оси должны быть подписаны, сверху должен располагаться соответствующий заголовок.
- Точки на графике функции $a = f_1(b)$ не должны иметь маркера, точки на графике функции $a = f_2(b)$ должны быть отмечены маркером, вид маркера в зависимости от варианта приведен в таблице 10 в конце задания. Вид линии графика функции $a = f_1(b)$ приведен в таблице 10.
 - Вид графика функции $z = f_3(x, y)$ приведен в таблице 10.
- Верхние оси координат должны иметь цвет, указанный в таблице 10, цвет нижних осей координат черный.
 - Цвет фона графиков должен быть белым.

- Остальные параметры графиков могут быть произвольными, но должны быть выбраны так, чтобы график был удобочитаем.
- Все выполняемые действия в файле сценария должны иметь соответствующие комментарии. Первый комментарий должен содержать номер лабораторной работы, номер варианта и фамилию автора или авторов.

5.2 Форма отчета

В качестве отчета по лабораторной работе представляется исходный файл сценария для математического пакета Scilab и файл графического формата с результатами построения. Результаты построения можно сохранять в файле графического формата вручную. Все файлы представляются в электронном виде. Оба файла должны удовлетворять условиям, описанным в п. 5.1, файл сценария должен корректно работать.

Все файлы должны быть помещены в отдельный каталог или архив.

При защите лабораторной работы могут быть заданы вопросы по любой строке файла сценария и его работе. При защите лабораторной работы должна быть продемонстрирована работоспособность файла сценария, в противном случае работа не принимается.

Пример выполнения лабораторной работы приведен ниже.

5.3 Пример выполнения

На рисунке 2.37 приведен результат выполнения задания для следующих условий:

$$f_1$$
: $a = \sin(b)$
 f_2 : $a = \cos(b)$, для $b \in \left[0; \frac{\pi}{4}\right]$, f_3 : $z = \frac{x}{y}$ для $\begin{cases} x \in [0; 1] \\ y \in [1; 2] \end{cases}$

Тип линий графика для функции f_1 — штрихпунктирная, маркер на графике функции f_2 — \square , цвет верхних осей координат — зеленый. Тип графика функции f_3 — контурный с десятью контурами и заливкой контуров цветом.

5.4 Критерии оценки

Следующие погрешности в выполнении работы приведут к снижению оценки за нее:

- отсутствие комментариев в файле сценария или их несоответствие тексту файла сценария;
- невыполненные требования к виду графика (цвет линии, тип маркера, расположение графиков и пр.);
- непонятные или нелогичные заголовки осей и графиков;
- неудачно подобранные параметры графика, приводящие к его неудобочитаемости.

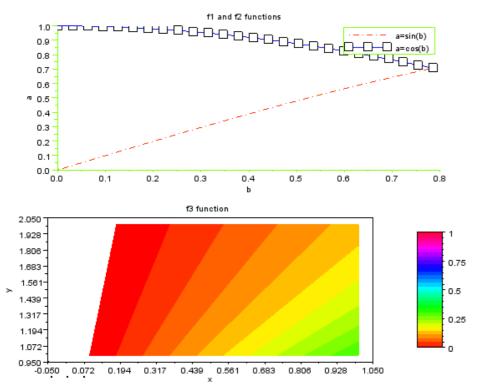


Рис 2.37 – Результат правильного выполнения задания

5.5 Варианты задания

Варианты задания приведены в таблицах 8 – 10.

Таблица 8. Задание по вариантам: функция f₃

Таблица 8. Задание по вариантам: функция 13					
№	2	X	y		f (v. v.)
варианта	min	max	min	max	f ₃ (x,y)
1	-4	4	-4	4	$x^3 \sin y + 1$
2	-4	4	-4	4	$x^2 \sin y - 1$
3	-4	4	-4	4	$(1+(y+5)^2)^{-1}+x^2$
4	-4	4	-4	4	$\sqrt{\left y^2+x^3\right }$
5	-2	2	-5	4	$\sqrt{\left y^3+x^2\right }$
6	-2	6	-5	6	$\cos y + xy$
7	-2	6	-5	6	$(1+(y+5)^2)^{-1}+xy$
8	0	1	0	π	$e^x + 3y$
9	0	1	0	π	$x^2 \cos y + 0.1$
10	-4	4	- π	π	$x^3 \cos y + 0.1$
11	0	2	0	π	$\cos(xy) - 0.5$
12	-4	4	-4	4	$e^{-y} + e^x - 2$
13	-4	4	-4	4	$e^{-y} - e^x - 0.1$
14	-1	1	-1	1	$e^{-xy}+1$
15	-4	4	-4	4	$\sqrt{y^2 + x^4}$
16	0	4	-4	4	$y\sqrt{(x+1)}$
17	-4	4	-4	4	$e^{y}(x+1)$
18	-4	4	-4	4	$\frac{x-3}{y+10}$
19	-4	4	-4	4	$2x + y^2$
20	-4	4	-4	4	$2x-y^3$
21	-4	4	-4	4	$e^x - 3y$
22	-4	4	-4	4	$\cos y - xy$
23	-1	1	-1	1	$-e^{-xy}+1$
24	-4	4	-4	4	$(y-1)\sqrt{ x+1 }$
25	0	1	0	1	$\frac{2x-3}{y+10}$

Таблица 9. Задание по вариантам: функции f₁ и f₂

1 ao.	Габлица 9. Задание по вариантам: функции f ₁ и f ₂							
$N_{\underline{0}}$	<u>6</u> b							
ва	min	max	$f_1(b)$	f ₂ (b)				
1	$\pi/2$	π	$\sin(b)\cdot\cos(b)\cdot b^{-2} + 5\cdot\ln(b)$	$\sin(5b)\cdot\cos(b)\cdot b^{-2} + 4\cdot\ln(b)$				
2	2,5	3	$\left(b^2 + 5b + \sqrt{b}\right)b^{-\frac{1}{3}} \cdot e^b$	$(b^2 + 5b + \sqrt{b})b^{-\frac{1}{3}} \cdot e^{\frac{b}{2}}$				
3	-1,99	-1,95	$\frac{1+\sin^2(8+b^3)}{\sqrt[3]{8+b^3}}$	$\frac{1+\sin^2(8+b^3)}{\sqrt[3]{9+b^3}}$				
4	5	10	$\left(1+\lg^2\frac{b}{4}\right)\left(10-e^{\frac{b}{4}}\right)$	$\left(1+\lg^2\frac{3b}{4}\right)\left(10-e^{\frac{b}{4}}\right)$				
5	5	20	$\sqrt[4]{ b^2-2.5 } + \sqrt[3]{\lg b^2}$	$\sqrt[4]{ b^2-2.5 } + \sqrt[3]{\lg(50b)^2}$				
6	0,25	0,5	$\sqrt[3]{10} \cdot (10^b - 2^b) \lg^{-1} 8$	$2\sqrt[3]{10} \cdot (10^b - 2^b) \log^{-1} 8$				
7	0,1	0,5	$\frac{\left(56 + \sin^2 5b\right)}{\left(\arccos\left(b^2\right) + e^{-b/2}\right)}$	$\frac{\left(56 + \left(2 + \sin 5b\right)^2\right)}{\left(\arccos\left(b^2\right) + e^{-b/2}\right)}$				
8	3	4	$\lg(b^2-1)\log_5^{-1}(8b^2-16)$	$\lg(b^2+b-1)\log_5^{-1}(8b^2-16)$				
9	0,1	0,15	$\frac{\arccos(b^2 - 0.5^2)}{\arcsin(b^2 - 0.2^2)}$	$\frac{(\arccos(b^2 - 0.5^2) + 5)}{\arcsin(b^2 - 0.2^2)}$				
10	2,5	3	$3^{b^2-1} - \lg(b^2-1) + \sqrt[3]{b^2-1}$	$3^{b^2} - \lg(b^2 - 1) + \sqrt[3]{b^2 - 1}$				
11	5	6	$\left(2\sqrt{b}-10\log_5 b\right)\lg^{-1} b-1 $	$(2\sqrt{b} - 10\log_5 b)\lg^{-1} 2b - 1 $				
12	11	20	$\sqrt{ 0.75 - 2b \cdot \lg^{-3} b}$	$\sqrt{ 0.75-2b \cdot\lg^{-3}(b+3)}$				
13	0,1	0,9	$\left(\arcsin^2 b + \arccos^4 b\right)^3$	$\left(\arcsin^2 b + \arccos^4 \left(\frac{b}{2}\right)\right)^3$				
14	$\pi/2$	$3\pi/2$	$\left(\sin^3 b + \cos^3 b\right) \ln b$	$\left(\sin^3(2b) + \cos^3 b\right) \ln b$				
15	10	15	$\lg^2(2+b)(2+b)^{-2}$	$\lg^2(2+b)(7+b)^{-2}$				
16	10	20	$(\sin b + \cos b) / \exp\left(b + \frac{3}{b}\right) \lg(b^2)$	$(\sin b + \cos(3b))/\exp\left(b + \frac{3}{b}\right)\lg(b^2)$				
17	21	23	$b\lg b + b / \left(\cos b - \frac{3}{b}\right)$	$b\lg(4b) + b\left(\cos b - \frac{3}{b}\right)$				
18	$2\pi/3$	π	$\sin^2(b-\pi)\sqrt{\cos(b+\pi)}/(b^2+5)$	$\sin^2(b-\pi)\sqrt{\cos(b+\pi)}/((b-2)^2+5)$				
19	0,1	0,3	$(\arcsin(2b) + \sin b)e^{2b} + 3$	$(\arcsin(2b) + \sin b)e^{2b} + 3b$				
20	5	6	$(\lg b + \ln b + e^b)(2\sin b + 1)$	$\left(\lg b + \ln b + e^b\right) \left(2\sin(3b) + 1\right)$				
21	0,1	0,5	$5\ln(b^2 + 2b - \arcsin(b))$	$5/(b+5)\ln(b^2+2b-\arcsin(b))$				
22	5	10	$(2b)^{-1}\sqrt{2b+\log_3(b)}$	$(2b+3)^{-1}\sqrt{2b+\log_3(b)}$				
23	$\pi/3$	π	$\sin(b)/(\sin(b+5)+10)+b\lg b$	$\sin(b)/(\sin(b+5)+10)+(b-5)\lg b$				
24	-1,5	-1,2	$\frac{\arcsin(b+2)}{\sqrt[3]{ b+5b-\arccos(b+2) }}$	$\frac{5 + \arcsin(b+2)}{\sqrt[3]{ b+5b-\arccos(b+2) }}$				
25	5	10	$(2b^5 - 10\log_3 b)\lg^{-1}(b+1)$	$(2b^5 - 10\log_3 b)(\lg(b+1) + 1)^{-1}$				

Таблица 10. Требования к оформлению графиков

<u>Гаолица 10. Греоования к оформлению графиков</u>						
тип линий f_1	маркер f ₂	цвет верхних осей координат	тип графика функции f ₃			
сплошная	•	черный	каркасный график без заливки поверхности цветом			
крупный штрих	+	зеленый	поверхность синего цвета без каркасных линий			
мелкий штрих	×	синий	поверхность зеленого цвета с каркасными линиями			
штрихпунктирная	0	голубой	поверхность, цвета которой соответствуют значению функции, без каркасных линий			
штрих – две точки	•	красный	каркасный график без заливки поверхности цветом			
пунктирная	<	пурпурный	контурный график с восемью контурами без заливки цветом			
двойной пунктир	Δ	желтый	контурный график с семью контурами с заливкой контуров цветом			
сплошная	∇	синий	каркасный график без заливки поверхности цветом			
крупный штрих	\oplus	зеленый	поверхность голубого цвета без каркасных линий			
мелкий штрих	0	голубой	поверхность красного цвета с каркасными линиями			
штрихпунктирная	*	красный	поверхность, цвета которой соответствуют значению функции, без каркасных линий			
штрих – две точки		пурпурный	поверхность желтого цвета без каркасных линий			
пунктирная	\triangleright	желтый	контурный график с девятью контурами без заливки цветом			
двойной пунктир	\triangleleft	черный	контурный график с шестью контурами с заливкой контуров цветом			
сплошная	☆	зеленый	каркасный график без заливки поверхности цветом			
крупный штрих	+	голубой	поверхность пурпурного цвета без каркасных линий			
мелкий штрих	×	красный	поверхность желтого цвета с каркасными линиями			
штрихпунктирная	⊕	пурпурный	поверхность, цвета которой соответствуют значению функции, без каркасных линий			
штрих – две точки	•	желтый	каркасный график без заливки поверхности цветом			
пунктирная	♦	черный	контурный график с семью контурами без заливки цветом			
	крупный штрих мелкий штрих штрихпунктирная штрих – две точки пунктирная двойной пунктир сплошная крупный штрих мелкий штрих штрихпунктирная штрих – две точки пунктирная сплошная штрих – две точки крупный штрих штрих – две точки пунктирная штрих – две точки пунктирная	тип линии трана прина пунктирная тип пунктирная трих пунктирная точки пунктирих трих трих пунктирная трих пунктирих трих трих пунктирих трих трих пунктирная трих трих трих трих трих трих трих трих	тип линии 11 f ₂ осей координат сплошная			

№ вар	тип линий ${ m f_1}$	маркер f ₂	цвет верхних осей координат	тип графика функции f ₃
21	двойной пунктир	Δ	зеленый	поверхность, цвета которой соответствуют значению функции, без каркасных линий
22	сплошная	∇	синий	контурный график с одиннадцатью контурами без заливки цветом
23	крупный штрих		голубой	контурный график с девятью контурами без заливки цветом
24	мелкий штрих	0	красный	контурный график с шестью контурами с заливкой контуров цветом
25	штрихпунктирная	*	пурпурный	поверхность пурпурного цвета с каркасными линиями

СПИСОК ЛИТЕРАТУРЫ

- 1. Квасов, Б. И. Численные методы анализа и линейной алгебры. Использование Matlab и Scilab : учебное пособие / Б. И. Квасов. Санкт-Петербург : Лань, 2021. 328 с. ISBN 978-5-8114-2019-3. Текст : электронный // Лань : электронно-библиотечная система. URL: https://e.lanbook.com/book/168887
- 2. Решение инженерных задач в среде Scilab : учебное пособие / А. Б. Андриевский, Б. Р. Андриевский, А. А. Капитонов, А. Л. Фрадков. Санкт-Петербург : НИУ ИТМО, 2013. 97 с. Текст : электронный // Лань : электронно-библиотечная система. URL: https://e.lanbook.com/book/71062
- 3. Математическое моделирование. Построение моделей и численная реализация [Электронный ресурс]: учебное пособие /Б.А. Горлач, В.Г. Шахов. 2-е изд., стер. Электрон. дан. СПб.: Лань, 2018. 292 с. Режим доступа: https://e.lanbook.com/book/103190. Загл. с экрана.
- 4. Уравнения математической физики: теория и практика [Электронный ресурс]: учебное пособие. /В.Г. Абдрахманов, Г.А. Булгакова 2-е изд., стер. Электрон. дан. М.: ФЛИНТА, 2019. 338 с. Режим доступа: https://e.lanbook.com/book/122548. Загл. с экрана

Захарова Виктория Юрьевна

Методические указания к выполнению лабораторных работ по курсу "Математическое моделирование физических процессов". Часть 1

Учебно-методическое пособие

В авторской редакции Редакционно-издательский отдел Университета ИТМО Зав. РИО Н.Ф. Гусарова Подписано к печати Заказ N2 Тираж Отпечатано на ризографе

Редакционно-издательский отдел

Университета ИТМО 197101, Санкт-Петербург, Кронверкский пр., 49, литер А