

Федеральное агентство по образованию

И.С. Тропин, О.И. Михайлова, А.В. Михайлов

**Численные и технические расчеты в среде Scilab
(ПО для решения задач
численных и технических вычислений)**

Учебное пособие

Москва 2008

Тропин И.С., Михайлова О.И., Михайлов А.В.

Т 742 Численные и технические расчеты в среде Scilab (ПО для решения задач численных и технических вычислений): Учебное пособие. — Москва: 2008. — 65 с.

Пособие содержит описание основных приемов работы с системой Scilab (v.4.1.2), предназначенной для численных научных и инженерных расчетов.

Целью данного пособия является описание возможностей Scilab в объеме, достаточном для использования системы при выполнении задач по математике и физике в школе.

Пособие ориентировано на преподавателей средней школы и не требует от читателя знания программирования. Возможности системы излагаются на примере решения школьных задач по математике и физике для 9—11 классов.

Глава 1. Оглавление

Оглавление.....	3
Введение.....	5
Глава 1. Начало работы.....	7
1.1. Начало и завершение сеанса.....	7
1.1. Описание меню — основного окна Scilab.....	10
1.2. Работа с командной строкой	11
Глава 2. Переменные в Scilab.....	13
2.1. Пользовательские переменные.....	13
2.2. Системные переменные.....	14
Глава 3. Математические операции в Scilab.....	15
Глава 4. Встроенные функции в Scilab.....	17
4.1. Встроенные математические функции.....	17
4.2. Функция вывода на экран <code>printf</code>	19
Глава 5. Функции, определяемые пользователем.....	22
Глава 6. Численное дифференцирование и интегрирование в Scilab.....	29
6.1. Интегрирование в Scilab.....	29
6.2. Вычисление производной в Scilab.....	30
Глава 7. Списки в Scilab.....	31
7.1. Создание списков. Обращение к элементам списков.....	31
7.1.1. Одномерные списки. Строки и столбцы.....	31
7.1.2. Многомерные списки. Таблицы.....	32
7.2. Математические действия над списком.....	34
Глава 8. Полиномы.....	37
8.1. Задание полиномов. Функция <code>poly</code>	37
8.2. Действия над полиномами.....	39
Глава 9. Решение уравнений.....	41
Глава 10. Решение систем линейных уравнений.....	45
Глава 11. Построение графиков.....	47
11.1. Функция <code>plot</code>	47
11.2. Оформление графиков.....	52
11.2.1. Изображение сетки в графической области.....	52
11.2.2. Вывод названий графика и осей	52
11.2.3. Нанесение описания линий.....	53
Глава 12. Примеры вычислений в Scilab.....	54
12.1. Вычисление площади круга и длины окружности.....	54
12.2. Вычисление длины вектора.....	54
12.3. Построение касательной к кривой в заданной точке.....	55
12.4. Решение уравнения графическим способом.....	57
12.5. Решение биквадратного уравнения.....	58
12.6. Вывод значения выражения на экран.....	58
12.7. Вычисление определенного интеграла.....	58
12.8. Вычисление длины гипотенузы.....	59
12.9. Задание функции для вычисления логарифма с произвольным основанием ..	59
Задания для самоконтроля.....	61
Глоссарий.....	63
Список литературы.....	65
Приложение А.....	66

Введение

Интенсификация научных исследований и инженерных разработок в условиях рыночных отношений, когда реализаций новых проектов ведется в жестких временных рамках, обусловила необходимость в программном обеспечении, позволяющем получать результат сложных математических задач в приемлемые сроки. В ответ на эту потребность различными фирмами и институтами был создан ряд программных продуктов, к их числу относятся, например, ANSYS, ROOT, MathLab и Scilab, рассматриваемый в данном пособии. Из числа упомянутых приложений ANSYS и MathLab являются коммерческими, ROOT и Scilab распространяются свободно.

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

Каждая из систем использует собственный язык программирования высокого уровня, за исключением ROOT, где применяется C++. Языки программирования каждой из упомянутых систем несовместимы между собой, однако программы, написанные на языке MathLab, могут быть импортированы в Scilab с помощью встроенных средств системы.

Обстоятельством, которое препятствует использованию этих систем в России, является отсутствие русскоязычного интерфейса и документации на русском языке. Для Scilab, например, авторам удалось обнаружить лишь три русскоязычные книги [1 — 3]. Система оснащена обширным встроенным справочником, однако информация в нем доступна лишь на английском и французском языках.

Целью данного пособия является описание возможностей Scilab в объеме, достаточном для использования системы при выполнении задач по математике и физике в школе. По мнению авторов, применение Scilab преподавателями в повседневной деятельности позволит увеличить эффективность их труда, а также преподнести материал более доходчиво, используя презентационные возможности системы. При исследовании функций, например, нетрудно написать небольшую программу, которая строит график функции при определенных параметрах. Можно дать ученикам возможность самим варьировать эти параметры, наблюдать эффект на графиках и делать выводы о влиянии коэффициен-

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab
тов на вид функции. Таким образом, вероятно, может быть организована лабораторная работа или демонстрация на уроке. Приведенный пример, пожалуй, самый очевидный способ использования Scilab в школе. Авторы надеются, что новые возможности, предоставляемые системой, позволят творческим преподавателям обнаружить и менее очевидные способы применения Scilab.

Глава 1. Начало работы

Scilab — это система компьютерной математики, предназначенная для решения вычислительных задач. Для того чтобы приступить к решению задачи, необходимо запустить Scilab. После запуска на экране появляется основное окно программы. Его появление означает начало сеанса работы пользователя. Закрытие окна означает прекращение сеанса. Далее показано, как начать и завершить сеанс, описаны составные части (области) основного окна, дано назначение пунктов меню. Приведенные сведения справедливы для версии Scilab 4.1.2, выпущенной в 2007 г.

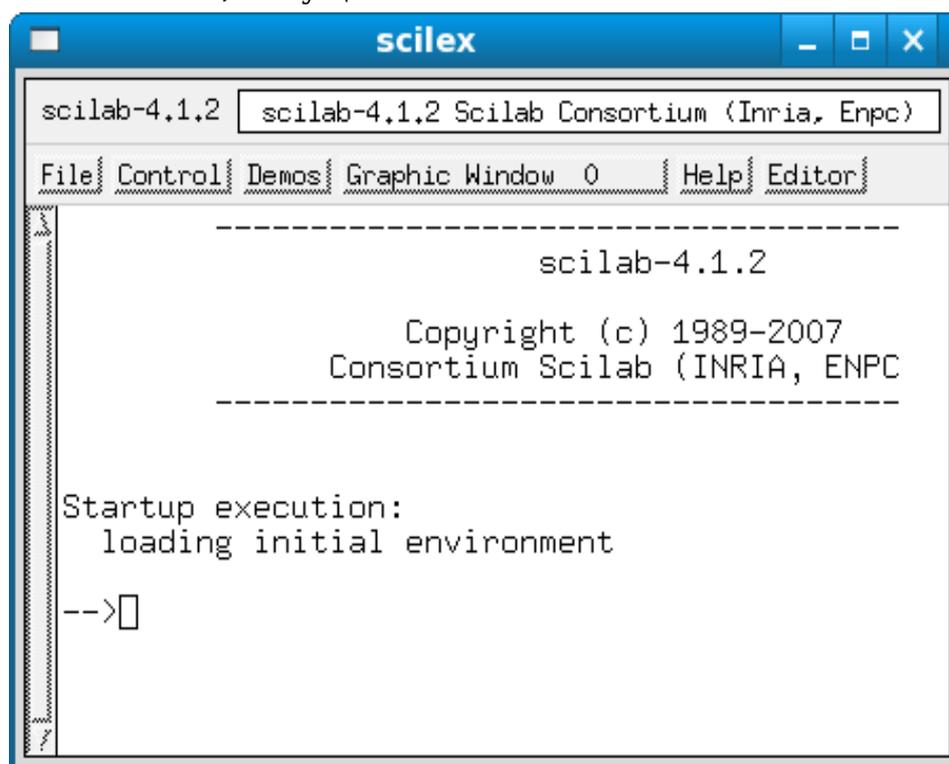


Рис. 1. Основное окно Scilab

1.1. Начало и завершение сеанса

Самым общим способом запуска является запуск программы через терминал. Для запуска программы введите в командной строке интерпретатора shell строку `scilab` и нажмите клавишу «Enter». После ввода команды откроется основное окно программы, показанное на рис. 1.

Можно вставить запуск Scilab в меню менеджера окон. Операция может быть произведена администратором системы или самим пользователем. В некоторых дистрибутивах Scilab присутствует в меню сразу после установки. На рис.2 представлен фрагмент системы меню KDE дистрибутива ПСПО.

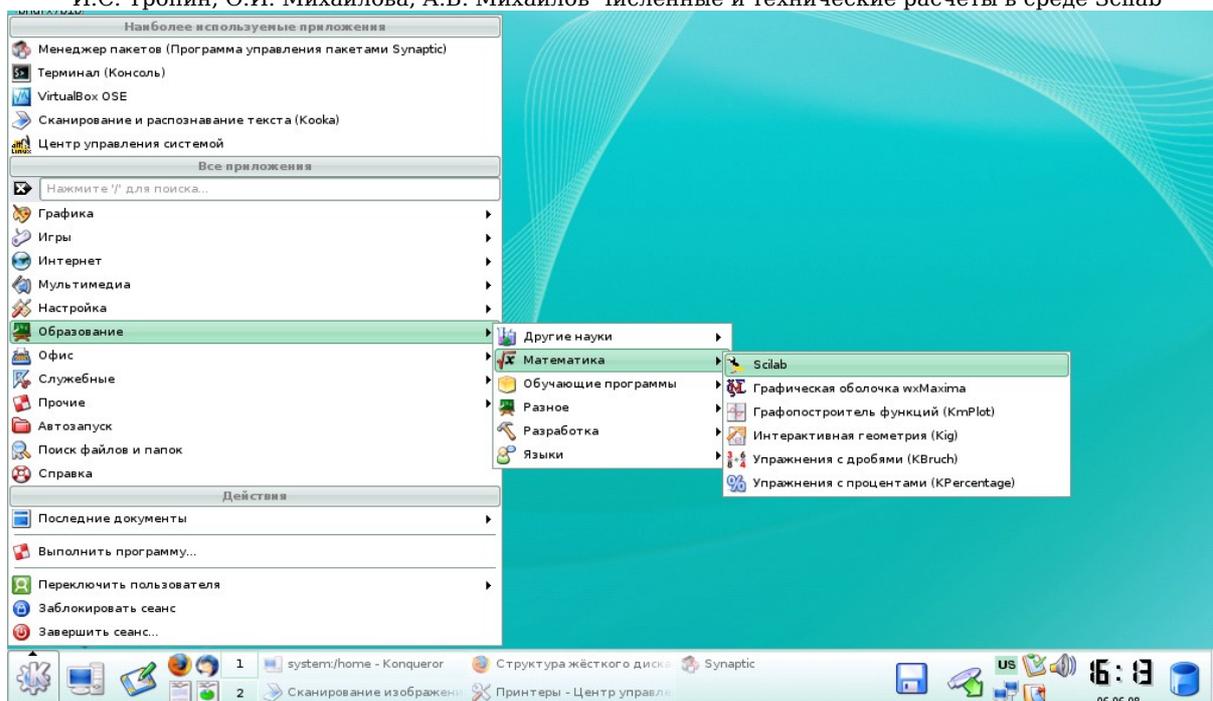


Рис. 2. Запуск Scilab в KDE

При использовании этого дистрибутива для старта Scilab в главном меню панели задач KDE выберите пункт «Образование», затем подпункт «Математика», во вкладке «Математика» выберите Scilab. Из меню менеджера окон XFCE запуск Scilab производится аналогично.

Основное окно Scilab (см. рис. 1) условно можно разделить на две области:

1. Область меню, которая расположена вверху экрана. Назначение пунктов меню описано в п. 1.1.
2. Рабочую область с командной строкой, в которой, собственно, и происходит решение задачи.

Индикатором готовности программы к работе является символ --> в рабочей области, возле которого находится курсор. В этом месте можно ввести команду Scilab и выполнить ее, нажав клавишу «Enter». Строку в рабочей области, где находится курсор, называют командной строкой.

Чтобы завершить сеанс, следует выполнить в командной строке команду quit. То же можно сделать, выбрав пункт Quit (Выход) в меню File (Файл) в верхней части основного окна. Способ использования меню и операции, которые можно выполнять с его помощью, рассмотрены далее.

1.1. Описание меню — основного окна Scilab

Для работы с выпадающим меню, которое находится в верхней части окна (см. рис. 1), нужно подвести курсор к одной из кнопок и нажатием на левую клавишу мыши открыть выпадающее меню. Удерживая клавишу мыши в нажатом состоянии, можно выбрать нужный пункт из появившегося списка. После того как пункт меню будет выделен, следует отпустить клавишу мыши.

Ниже приведено описание операций, которые доступны с использованием графического интерфейса основного окна, рядом с английскими названиями кнопок и пунктов меню в скобках приведен русский перевод:

- File (Файл) — операции с файлами.
 - File Operations (Операции) — открывает меню для работы с файлами.
 - Kill (Убить) — закрытие программы при любых условиях, например, если идут расчеты.
 - Quit (Выход) — выход из Scilab.
- Control (Управление) — управление выполнением программы.
 - Resume (Возобновить) — возобновляет приостановленный расчет.
 - Abort (Прекратить) — прерывает выполнение программы.
 - Stop (Останов) — приостанавливает выполнение программы.
- Demos (Демонстрации) — открывает окно с перечнем примеров, демонстрирующих возможности Scilab. Выбор какого-либо пункта из списка запускает одно из демонстрационных приложений. Их исходные тексты находятся в директории `/usr/lib/scilab-4.1.2/demos/`
- Graphic Window (Графическое окно) — операции над графическими окнами. Графические окна используются в Scilab, например, для построения графиков. Каждое графическое окно имеет номер. Нумерация окон начинается с нуля. В кнопке меню отображается номер текущего окна, к которому применяются операции из выпадающего меню:
 - Set (create) window (Установить (создать) окно) — установить окно в качестве текущего, если окно с текущим номером отсутствует, оно будет создано.
 - Raise (create) window (Развернуть (создать) окно) — развернуть окно с текущим номером, если окно отсутствует, оно будет создано.
 - Delete graphic window (Удалить графическое окно) — закрытие текущего графического окна.
 - + — Увеличивает номер текущего графического окна на единицу.
 - «-» — Уменьшает номер текущего окна на единицу.
- Help (Помощь) — справка по программе Scilab.

- Help browser (Справочник)— справочная система, которая содержит полное справочное руководство по использованию всех функций Scilab. К сожалению, в настоящее время справочная информация доступна только на английском и французском языках. Справочник снабжен системой поиска. Поиск ведется по ключевому слову в названии или определении функции.
- Argopros (что касается, по поводу) — поиск ключевых в справочной системе Scilab. В отличие от Help Browser поиск ведется еще и во всем тексте справки (расширенный поиск похожих слов).
- Configure (Настройка) — выбор программы, с помощью которой открывается справка. По умолчанию применяется Scilab Browse.
- Editor (Редактор) — вызов встроенного редактора Scilab, который может быть использован при программировании приложений.

1.2. Работа с командной строкой

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

Текст в командной строке можно редактировать с помощью стандартных клавиш редактирования текста:

- клавиша Backspace — удаление текста перед курсором;
- клавиша Delete — удаление текста после курсора.

Текст в области просмотра редактировать нельзя, но, если есть необходимость повторить команду, то при помощи клавиши управления курсором ↑ ранее выполненные команды могут быть возвращены в командную строку. После этого их можно редактировать и выполнять повторно. При нажатии клавиши ↓ команды появляются в командной строке в обратном порядке.

Список команд, хранимых системой, не ограничивается текущим сеансом, однако при необходимости текст, введенный в текущем сеансе, можно сохранить и в виде текстового файла. Для этого необходимо в командной строке набрать **diary ('name')**, после чего файл с именем name появится в рабочей директории.

Если текст, который необходимо поместить в командную строку, находится в другом окне, то, для начала, выделите его мышью, затем перейдите в основное окно Scilab и нажмите среднюю кнопку мыши. Выделенный текст будет помещен в командную строку, начиная с того места, где находился текстовый курсор. В случае, если мышь имеет две кнопки, то для вставки попробуйте использовать одновременное нажатие обеих кнопок.

Глава 2. Переменные в Scilab

2.1. Пользовательские переменные

В рабочей области Scilab можно определять переменные для дальнейшего их использования в различных выражениях. Определить переменную значит присвоить ей какое-либо значение. Оператором присваивания в Scilab является «=». Процедура присваивания оформляется следующим образом: **имя переменной=значение переменной**. Каждая переменная перед использованием в выражении должна быть определена. В противном случае Scilab выдаст предупреждение об ошибке. Переменным могут быть присвоены не только численные значения, например переменная может быть определена как строка символов (Листинг 1). При определении переменной ей может быть присвоено значение результата вычисления. Тогда в результате выполнения команды на экран будет выведено: **имя переменной=результат вычисления** (Листинг 2).

Листинг 2

```
-->d='stroka'
d =
stroka
```

Листинг 1

```
--> d=6+8
d =
14.
```

П р и м е ч а н и е . Если в течение одного сеанса присвоить некоторое значение ранее определенной в этом сеансе переменной, то в дальнейшем именно это значение будет использоваться программой во всех вычислениях, содержащих переменную. Говорят, что переменная переопределена.

Задания для самоконтроля:

1. Задайте пользовательские переменные для хранения числа 5, строки file.
2. Задайте пользовательскую переменную j для хранения результата деления: $\frac{1}{3}$.
3. Определите переменные для хранения двух произвольных чисел, затем найдите разность этих чисел используя в математическом выражении переменные.

2.2. Системные переменные

При выполнении операции Scilab обязательно присваивает ее результат какой-либо переменной, если в командной строке нет оператора присваивания, то результат будет присвоен системной переменной с именем `ans` (от англ. `answer` — ответ) (Листинг 3). Переменную `ans` можно использовать в последующих вычислениях, но ее значение будет изменяться каждый раз после выполнения команды без оператора присваивания.

Листинг 3

```
-->3+4
ans =
7.
```

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

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

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

Задания для самоконтроля:

1. Найдите длину окружности, имеющей радиус 5 см.
2. Найдите сумму двух произвольных чисел, затем вычтите 1 из суммы, используя переменную `ans`.

Глава 3. Математические операции в Scilab

Для выполнения простейших арифметических операций Scilab использует следующие операторы:

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

Чтобы вычислить значение арифметического выражения, необходимо ввести его в командную строку и нажать «Enter». Результат вычисления появится в рабочей области. Пример вычисления выражения $4 \times 8 - \frac{45.6}{5.67} + 2^3$ демонстрирует Листинг 4. При вводе выражения применена операция деления слева «\». Поскольку значение выражения не присвоено пользовательской переменной, то результат присваивается системной переменной ans:

Листинг 4

```
-->4*8-5.67\45.6+2^3
ans =
  31.957672
```

Если вычисляемое выражение длинное и желательно перенести его запись на следующую строку, то в конце незавершенной строки необходимо ввести три (или более) точки. После этого можно нажать «Enter» и продолжать набор оставшейся части на следующей строке (Листинг 5).

Листинг 5

```
-->1+3+5+7+8...
-->+4+8+9...
-->+10+11+12
ans =
  78.
```

Если после ввода команды нажать клавишу «Enter», то в рабочей области появится результат выполнения этой команды (Листинг 7). Если результат отображать не нужно, то набор команды следует завершить символом «;» (Листинг 6).

Листинг 6

-->z=1; Результат выполнения операции (присваивания значения) **не** отображается в рабочей области.

Листинг 7

-->z=1 Результат выполнение операции (присваивания значения)
z = (присваивания значения) отображается в рабочей области.

1.

Задания для самоконтроля:

1. Вычислите значения выражения $5^3 + \frac{4}{3} - 6 * 8$.
2. Определите, какое из чисел больше, $\frac{5}{139}$ или $\frac{9}{901}$.
3. Вычислите значение выражения $\frac{153}{5 + 64 * (\frac{1}{33} - 7)^3 + \frac{7}{8}}$, используя деление слева.

Глава 4. Встроенные функции в Scilab

4.1. Встроенные математические функции

Система Scilab имеет достаточный набор встроенных функций. В него входят:

Таблица 4.1

Встроенные функции Scilab

Имя функции	Возвращаемое значение	Область определения x	Диапазон возвращаемых значений
1	2	3	4
$\sin(x)$	Синус угла x	$(-\infty; +\infty)$, x задается в радианах.	$[-1; +1]$
$\cos(x)$	Косинус угла x	$(-\infty; +\infty)$, x задается в радианах.	$[-1; +1]$
$\tan(x)$	Тангенс угла x	$\left(\frac{-\pi}{2} \pm \pi n; \frac{\pi}{2} \pm \pi n\right)$, $n = 0, \pm 1, \pm 2, \dots$, x задается в радианах.	$(-\infty; +\infty)$
$\cotg(x)$	Котангенс x	$x \neq \pi n$, $n = 0, \pm 1, \pm 2, \dots$, x задается в радианах.	$(-\infty; +\infty)$
$\text{asin}(x)$	Арксинус x	$[-1; +1]$	$\left[\frac{-\pi}{2}; \frac{\pi}{2}\right]$, результат — в радианах
$\text{acos}(x)$	Арккосинус x	$[-1; +1]$	$[0; \pi]$, результат — в радианах
$\text{atan}(x)$	Арктангенс x	$(-\infty; +\infty)$	$\left(\frac{-\pi}{2}; \frac{\pi}{2}\right)$, результат — в радианах
$\exp(x)$	Число e , возведенное в степень x	$(-\infty; +\infty)$	$(0; +\infty)$

1	2	3	4
$\log(x)$	Натуральный логарифм x	$(0; +\infty)$	$(-\infty; +\infty)$
$\log_2(x)$	Логарифм x по основанию 2	$(0; +\infty)$	$(-\infty; +\infty)$
$\log_{10}(x)$	Десятичный логарифм x	$(0; +\infty)$	$(-\infty; +\infty)$
$\text{sqrt}(x)$	Квадратный корень из x	$[0; +\infty)$	$[0; +\infty)$

Для примера вычислим значение выражения $c = \sqrt{\left| \cos\left(\frac{y}{x}\right) \right| + \log_{10}(e^{y^x})}$ при $x=4,5$, $y=8,09$.

Листинг 8

```
-->x=-4.5;
-->y=8.09;
-->c=sqrt(abs(cos(y/x)))+log10(exp(y^x))
с =
0.4744167
```

В приведенной выше таблице области определения функций соответствуют правилам, принятым в школе. Однако Scilab способен вычислить, например, квадратный корень или логарифм и для отрицательных чисел. В этом случае ответом будет так называемое комплексное число. Если попытаться вычислить, например, $\lg(-10)$, то Scilab выдаст в качестве ответа $1.+1.3643764i$. Поскольку комплексный анализ в школе не изучается, то получение ответов вида $a+bi$, где a и b – некоторые числа, будет означать ошибку либо в условии задачи, либо в ходе ее решения.

Несколько необычным, пожалуй, может показаться поведение функции $\text{atan}(x)$. В качестве ответа эта функция выдает углы только из первого и четвертого квадранта. Если требуется получить значение угла в правильном квадранте, следует использовать разновидность функции арктангенса вида $\text{atan}(y, x)$, где y – синус угла или длина противолежащего катета в прямоугольном треугольнике, x – косинус угла или длина прилежащего катета. Область значений этой функции находится в диапазоне $(-\pi; \pi]$.

Задание для самоконтроля:

1. Вычислите значение выражения $\sqrt{\sin(e^6) + \ln(7)}$.
2. Вычислите значение выражения $\arctg^3(\frac{\sqrt{3}}{3})$.
3. Вычислите значение выражения $\log \sqrt[3]{e^{7\pi}}$.

4.2. Функция вывода на экран `printf`

Функция **printf** служит для вывода на экран значений переменных и текстовых комментариев. Формат вывода определяется аргументами функции. Назначение и использование **printf** в Scilab полностью аналогично использованию одноименной функции в языке C. Читатель, знакомый с языком C, может пропустить этот раздел.

Обращение к **printf** выглядит следующим образом:

printf("текст для вывода %символ1 %символ 2 ... %символn", переменная1, переменная2, ..., переменнаяn)

В результате обращения к функции **printf** на экран вводится текст, указанный в кавычках, и значения переменных (**переменная1, переменная2, ..., переменнаяn**), на месте соответствующих символов (**%символ1 %символ2 ... %символn**), определяющих формат вывода. Значение переменной будет находиться в той части выведенного текста, где находится соответствующий ей символ. Символ всегда ставится после знака **%**. При вызове функции **printf** используются символы **s** и **f**. Символ **s** используется, если переменная, значение которой необходимо вывести, содержит строку символов. Если переменная содержит число, используется символ **f**. Запись **%.nf** означает, что число должно быть выведено с *n* знаками после запятой. Текст, содержащийся в кавычках, выводится без изменений, а на месте символов выводятся значения соответствующих переменных. Текст, содержащийся в кавычках, выводится в виде одной строки. Если есть необходимость перенести часть текста на следующую строку, то перед этим участком текста необходимо ввести символ **\n**. Это неотображаемая «буква», которая вызывает перемещение курсора в начало следующей строки.

Рассмотрим примеры использования **printf**.

- Зададим переменную *h*, в которой будет храниться строка символов `variable` и выведем значение переменной на экран при помощи функции `printf` (Листинг 9).

Листинг 9

```
-->h='variable'
h =
variable
-->printf("text: %s",h)
text: variable
```

В данном случае первая строка определяет переменную h для хранения строки символов — `variable`. При вызове функции `printf` текстом для ввода является запись «text:». Символ **s** в кавычках означает, что на его месте будет выведено значение переменной, содержащей строку символов. После кавычек указана переменная — h , именно ее значение выведено в тексте на месте `%s`.

- Выведем на экран значения числовых переменных в сочетании с текстом (Листинг 10).

Листинг 10

```
-->d=1.15*%pi
d =
3.6128316
-->printf("znachenie d - %.5f \n e=%.7f" ,d,
%e)
znachenie d - 3.61283
e=2.7182818
```

При вызове функции **printf** текстом для вывода является запись: *znachenie d — e=*, она будет выведена без изменений. В кавычках содержатся символы **%.5f** и **%.7f**, это означает, что на их месте будут выведены значения числовых переменных с пятью и семью знаками после запятой соответственно. После кавычек через запятую указаны имена переменных, их значения будут выведены в тексте на месте символов **%.5f** и **%.7f**. Переменная d указана первой, соответственно ее значение будет выведено на месте первого символа — **%.5f**, а значение переменной e — на месте второго. В кавычках также присутствует символ **\n**, как видно, текст, стоящий после него, перенесен на следующую строку.

Примечание: Если необходимо вывести на экран значение математического выражения, то это выражение можно поставить после кавычек вместо имени переменной. Значение выражения будет вычислено и выведено на месте соответствующего символа (Листинг 11). При этом значение результата вычисления не будет присвоено никакой переменной.

Листинг 11

```
-->printf("%.5f",1/3)  
0.33333
```

Символ `%.5f` в кавычках означает, что на его месте на экран предполагается вывести число с пятью знаками после запятой. После кавычек указано выражение — деление $\frac{1}{3}$. Его значение вычисляется и выводится на экран на месте соответствующего символа. При этом значение вычисления не присваивается никакой переменной.

Задания для самоконтроля:

Выведите на экран:

1. Текст: *znaczenie funkcji*.
2. Значение числа $\frac{1}{9}$ с тремя знаками после запятой.
3. Значение переменной h , предварительно присвоив ей значение $\sin\left(\frac{\pi}{7}\right)$.

Глава 5. Функции, определяемые пользователем

В виде пользовательских функций обычно реализуют последовательность команд, которая требует многократного выполнения. Ниже будут рассмотрены два способа создания функций в Scilab.

Рассмотрим, для начала, описание функций с использованием оператора **deff**. В качестве примера используем **deff** для определения функции двух переменных:

$$f(x, y) = \sqrt{\left| \cos\left(\frac{y}{x}\right) \right|} + \log_{10}(e^{y^x}).$$

Для вычисления значения этой функции необходимо задавать пары значений x и y . В программировании говорят, что функция имеет два **входных** параметра. В математике значением функции является число, в программировании это число должно быть присвоено переменной, определенной внутри функции. Такую переменную называют **выходным** параметром.

Значение выходного параметра в программировании называют **возвращаемым значением**. В данном случае функция должна возвращать одно значение, для этого должен быть предусмотрен один выходной параметр.

Заметим, что в Scilab функция может возвращать более одного значения. В этом случае каждому возвращаемому значению должен быть поставлен в соответствие выходной параметр. Такие функции в программировании называют подпрограммами. Однако, программисты часто используют термины «функция» и «подпрограмма» как синонимы.

Приступая к разработке функции, необходимо определить количество входных и выходных параметров, а также придумать для функции имя. Назовем создаваемую функцию *fun*. Рассмотрим теперь синтаксис оператора **deff**, которому необходимо сообщить накопленную информацию о функции. Его общее выражение может быть представлено следующим образом:

deff(строка1, строка2).

Отсюда видно, что **deff** имеет два параметра, и эти параметры являются строками символов. По правилам Scilab первый параметр **строка1** несет в себе информацию о входных и выходных параметрах, а так же, об имени функции. В нашем случае **строка1** может быть записана следующим образом:

'с=fun(x,y)',

где s – переменная, которая представляет собой выходной параметр; *fun* – имя функции; две переменных в круглых скобках, разделенные запятыми, задают имена входных параметров. Во втором параметре **строка2** следует написать выражение для функции и присвоить вычисленное значение выходному параметру. При этом должны использоваться

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab
имена переменных, определенные в переменной **строка1**. В соответствии с этими правилами, второй параметр запишется в виде

$$'c=\sqrt{\text{abs}(\cos(y/x))}+\log_{10}(\exp(y^x))'$$

В этой строке выходной параметр, названный c , в результате операции присваивания получит значение, вычисленное по заданной формуле. При вычислении используются входные параметры с именами x и y , указанные в параметре **строка1**.

Листинг 12 демонстрирует определение функции fun с помощью оператора **deff** и использование fun для вычисления значения при $x=-4.5$ и $y=8.09$.

Листинг 12

```
-->deff('c=fun(x,y)', 'c=sqrt(abs(cos(y/x)))+log10(exp(y^x))');  
-->a=-4.5; b=8.09; g=fun(a,b)  
g =  
0.4744167
```

Первая строка листинга содержит определение функции с помощью оператора **deff**. Во второй строке производится присваивание значения -4.5 и 8.09 переменным a и b соответственно. Причем, результаты выполнения этих двух операций не отображаются на экране, поскольку запись каждой из них заканчивается символом «;». Далее, переменной g присваивается результат выполнения функции fun . Первым параметром при вызове функции указано значение переменной a , вторым – значение переменной b .

Переменные, которые используются при вызове функции, в данном случае a и b , называют фактическими параметрами. Входные параметры x и y , определенные внутри функции, называют формальными параметрами. При вызове функции в Scilab происходит неявное присваивание формальным параметрам значений параметров фактических.

Способ передачи входных и выходных параметров при вызове функций с использованием неявного присваивания в программировании называют передачей по значению.

После того как значение переменной g будет определено, оно отображается на экране, поскольку завершающий символ «;» в последнем выражении отсутствует.

При вызове функции в качестве фактических аргументов, можно сразу указывать числовые значения, так что предыдущий фрагмент можно записать короче:

Листинг 13

```
-->deff('c=fun(x,y)', 'c=sqrt(abs(cos(y/x)))+log10(exp(y^x))');
-->fun(-4.5,8.09)
ans =
    0.4744167
```

Как видно, упрощение программы не повлияло на ответ, он остался тем же, что и на предыдущем листинге.

Общий вид оператора **deff** для определения функции с несколькими выходными параметрами, приведен в Приложении А.

Другой способ задания функции — использование служебного слова `function`. Преимущество этого способа заключается в том, что размер описания функции не ограничен одной строкой. Рассмотрим простой пример создания функции, вычисляющей значение квадрата аргумента.

Следуя рецепту, приведенному выше, определяем, что функция имеет один входной и один выходной параметр, дадим функции имя f . Описание функции приведено в Листинге 14.

Листинг 14

```
-->function c=f(x),
-->c=x^2
-->endfunction
-->f(2)
ans =
    4.
```

Из листинга видно, что описание функции начинается со служебного слова `function` (функция) и завершается словом **endfunction** (конец функции). Первое выражение, следующее за **function**, является полным аналогом параметра **строка1** в операторе **deff**, с той лишь разницей, что выражение не требуется заключать в кавычки. Так что запись $c=f(x)$ означает, что выходной параметр у функции один, и он имеет имя c , функция называется f , и она имеет один формальный входной параметр x .

Следующее выражение после запятой, говорит о том, что выходному параметру следует присвоить значение входного параметра x , возведенное в квадрат.

При задании функции вышеописанными способами функция является внутренней функцией текущего сеанса. Если она понадобится в следующем сеансе, ее придется описать снова.

Способы, рассматриваемые ниже, позволяют задавать внешние функции, которые могут быть использованы в любом сеансе. Эти прие-

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab
мы окажутся особенно полезными, если создаваемая функция будет часто использоваться в работе.

Отличия в подготовке внешних функций начинаются с того, что определение функции нужно набирать не в командной строке, а в текстовом редакторе, например встроенном редакторе Scilab, называемом SciPad.

Набранный текст необходимо сохранить в файле с произвольным именем и расширением **sce**. SciPad предлагает это расширение по умолчанию. Теперь, для того чтобы использовать функцию в сеансе можно загрузить ее из файла. Это можно сделать следующими способами:

1. Выполнить в командной строке команду `exec`. Ее единственным аргументом является строка с указанием положения сохраненного файла в файловой системе. Например, если файл `f.sce` был сохранен в домашнем каталоге пользователя, то команда загрузки будет выглядеть следующим образом: `exec('~'/f.sce')`. После этого функция, описанная в файле, может быть вызвана в текущей сессии. Листинг 15 демонстрирует еще один пример загрузки подпрограммы.

Листинг 15

```
-->exec('~'/pr/RZL.sce');
-->RAZL(5,1,4)
ans =
polinom ne imeet nulei
-->RAZL(5,6,1)
1.0 x+6.0 x+5.0=1.0(x-(01.0))(x-(-5.0)) ans =
polinom razlozen na mnojiteli
```

В этом примере загрузка производится из файла `RZL.sce`, находящегося в подкаталоге `pr` пользовательского каталога. Текст подпрограммы `RAZL`, реализующей разложение на множители многочлена второй степени, приведен в Листинге 16. Реализация функции `RAZL` будет обсуждаться в конце раздела.

2. Использовать графический интерфейс редактора SciPad. Для этого нужно запустить редактор SciPad, нажав мышью кнопку `Edit` (Редактор) в верхней части основного окна Scilab, открыть в редакторе текстовый файл, содержащий подпрограмму. Затем в меню SciPad нажать мышью кнопку `Execute` (Выполнить) и в выпадающем меню выбрать пункт `Load into Scilab` (Загрузить в Scilab). В результате текст программы, отображающийся в окне редактора, будет выполнен в основном окне. С этого момента функция может использоваться в текущем сеансе.

Во время сеанса, последовательно выполняя команды в командной строке, можно видеть результат выполнения каждой введенной ко-

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab манды. В зависимости от результата можно планировать ту или иную последовательность действий.

После обращения к функции ситуация меняется, вычисления внутри нее производятся неинтерактивно. Это означает, что нет способа видеть значение каждого выражения и на этой основе динамически формировать процесс вычислений. Между тем часто алгоритм вычисления зависит от значения одного или нескольких производных параметров, возникающих в результате операций над входными параметрами функции.

Таким образом, при написании функций необходим программный инструмент, который позволял бы анализировать значения переменных (выражений) и априорно задавать ход вычислений в зависимости от их возможных значений.

Рассмотрим, например, задачу о разложении на множители полинома второй степени вида

$$p_2(x) = a_0 + a_1 x + a_2 x^2.$$

И зададимся целью написать такую функцию в Scilab. Пусть функция в качестве входных параметров получает коэффициенты a_0 , a_1 и a_2 . Ее задачей будет печатать представления полинома на экране в виде.

$$a_0 + a_1 x + a_2 x^2 = a_2 (x - x_1)(x - x_2),$$

где x_1 , x_2 — нули полинома.

Разложение на множители требует определения нулей полинома, которые находятся решением квадратного уравнения. Существование и количество корней такого уравнения в школьной программе ставятся в зависимость от значения дискриминанта. Это означает, что внутри функции нужно вычислить дискриминант D , затем, основываясь на его значении, принять решение о дальнейших действиях. Если окажется, что $D \geq 0$, то следует вычислить корни и напечатать результат разложения полинома на множители на экране. В противном случае ($D < 0$) необходимо вывести сообщение об отсутствии решения.

Пример реализации функции разложения на множители квадратного полинома приведен в Листинге 16.

Листинг 16

```
1. function d=RAZL(a0,a1,a2)
2.   printf("\n%.1f x^2+(%.1f) x+(%.1f)", a2,a1,a0);
3.   D=a1^2-4*a0*a2;
4.   if D>=0 then
5.     x1=(-a1+sqrt(D))/(2*a2);
6.     x2=(-a1-sqrt(D))/(2*a2);
7.     d='polinom razlojen na mnojiteli';
8.     printf("=%.1f(x-(%.1f))(x-(%.1f))",a2,x1,x2);
9.   else d='polinom ne imeet nulei';
10. end
endfunction
```

Строки программы пронумерованы, но эти номера не являются частью программы и приведены для удобства чтения.

Из первой строки программы можно видеть, что функция имеет один выходной параметр, названный d . Функция названа *RAZL*. Функция имеет три формальных входных параметра с именами a_0 , a_1 , a_2 .

В строке 2 с использованием функции `printf` на экран выводится исходное выражение полинома.

В строке 3 переменной D присвоено значение дискриминанта квадратного уравнения, вычисленное по значениям формальных параметров.

Теперь ход дальнейших вычислений зависит от значения D . Эта зависимость реализована в конструкции условного выполнения `if - then - else - end`, которая занимает строки 4 — 10.

Для выяснения семантики этого выражения приведем тот же фрагмент программы, заменив английские служебные слова `if-then... else... end` их русским переводом (Листинг 17).

Листинг 17

```
...
4.  ЕСЛИ D>=0 ТО
5.    x1=(-a1+sqrt(D))/(2*a2);
6.    x2=(-a1-sqrt(D))/(2*a2);
7.    d='polinom razlojen na mnojiteli';
8.    printf("=%.1f(x-(%.1f))(x-(%.1f))",a2,x1,x2);
9.  ИНАЧЕ d='polinom ne imeet nulei';
10. КОНЕЦ
...
```

При такой записи легко видеть, что после слова `if` (ЕСЛИ) по смыслу должно следовать условие.

Операция отношения «больше или равно» в Scilab записывается двумя символами `>=`, так что в данном примере условие имеет вид `D>=0`.

Если условие окажется истинным, выполняется ветвь `then` (ТО). В данном примере она содержит вычисление двух корней уравнения, их значения присваиваются переменным `x1` и `x2` (строки 5 и 6). В строке 7 выходной параметр функции `d` получит строковое значение `'polinom razlojen na mnojiteli'`. Функция `printf` в строке 8 напечатает символ «=`»` и, затем, полином в виде $a_2*(x-x_1)*(x-x_2)$.

На этом ветвь `then` (ТО) завершится и выполнение программы продолжится с первого оператора после слова `end` (КОНЕЦ). В данном случае функция завершит свою работу, значение строковой переменной `d` будет передано вызывающей программе.

При невыполнении условия, следующего за оператором `if` (ЕСЛИ), выполнение программы переходит в ветвь `else` (ИНАЧЕ). В этой ветви корни уравнения не вычисляются. Переменной `d` присваивается строка `'polinom ne imeet nulei'`.

Описание альтернативных ветвей выполнения программы завершается словом `end` (КОНЕЦ).

Пример выполнения функции `RAZL` представлен в Листинге 15.

Задания для самоконтроля:

1. Задайте пользовательскую функцию для вычисления x^n , входными параметрами функции должны являться значение числа x значение числа n .
2. Задайте пользовательскую функцию для вычисления проекций вектора по его длине и углу между направлением вектора и положительным направлением оси Ox . Входными параметрами функции должны являться значение длины вектора, угол между направлением вектора и положительным направлением оси Ox .

Глава 6. Численное дифференцирование и интегрирование в Scilab

6.1. Интегрирование в Scilab

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

Рассмотрим использование функции **int** на примере вычисления определенного интеграла:

$$\int_0^{\pi} 3 \cos \frac{x}{2} dx$$

В данном примере $a=0$; $b=\pi$; $f(x)=3\cos(\frac{x}{2})$.

Для вычисления интеграла в Scilab необходимо сначала задать функцию $f(x)$ (см. п. 5) задание функции отражено в Листинге 18.

Листинг 18

```
-->function y=f(x),y=3*cos(x/2) endfunction
```

После этого можно вычислить интеграл (Листинг 19).

Листинг 19

```
-->intg(0,%pi,f)
```

```
ans =
```

```
6.
```

Для проверки полученного результата вычислим интеграл аналитически:

$$\int_0^{\pi} 3 \cos \frac{x}{2} dx = 3 * 2 * \sin \frac{x}{2} \Big|_0^{\pi} = 6 * \sin \frac{\pi}{2} - 6 * \sin \frac{0}{2} = 6 * 1 - 0 = 6 .$$

Результаты аналитического и численного (Scilab) вычислений совпадают.

Задания для самоконтроля:

1. Найдите работу по перемещению тела на расстояние в 30 см под

действием силы; $F(x) = \frac{5+x^2}{x-1}$.

2. Вычислите интеграл $\int_2^5 (-\sin \pi x) dx$.

3. Вычислите площадь фигуры, ограниченной линиями

$y = \left(\sin \frac{x}{4} + \cos \frac{x}{4} \right)^2$ и $y = \left(\sin \frac{x}{4} + \cos \frac{x}{4} \right)^2$.

6.2. Вычисление производной в Scilab

В Scilab можно вычислять производную функции в заданной точке. Вычисление происходит при помощи команды **numdiff(f,x₀)**, где f — имя дифференцируемой функции переменной x , x_0 — координата точки в которой необходимо вычислить производную. Рассмотрим пример применения команды **numdiff**. Вычислим производную функции

$$f(x) = \frac{16}{\sqrt{x^2 - 5x + 8}} \text{ в точке } x_0 = 4.$$

Аналитическое решение:

$$f'(x) = \left(\frac{16}{\sqrt{x^2 - 5x + 8}} \right)' = -\frac{16}{(\sqrt{x^2 - 5x + 8})^2} * \frac{1}{2\sqrt{x^2 - 5x + 8}} * (2x - 5) = \dots$$

$$= -\frac{16 * (2x - 5)}{x^2 - 5x + 8} * \frac{1}{2\sqrt{x^2 - 5x + 8}} = \frac{16 * (5 - 2x)}{x^2 - 5x + 8} * \frac{1}{2\sqrt{x^2 - 5x + 8}}$$

$$f'(4) = \frac{16 * (5 - 2 * 4)}{4^2 - 5 * 4 + 8} * \frac{1}{2\sqrt{4^2 - 5 * 4 + 8}} = \frac{-3 * 16}{16} = -3.$$

Решение в Scilab:

Зададим дифференцируемую функцию (Листинг 20):

Листинг 20

```
-->function l=f(x),l=16/(sqrt(x^2-5*x+8)) endfunction
```

После чего, воспользовавшись командой **numdiff**, вычислить производную функции в точке $x_0 = 4$ (Листинг 21):

Листинг 21

```
-->numdiff(f,4)
ans =
-3.0000001
```

В приведенном примере аргументами **numdiff** являются f — имя дифференцируемой функции переменной x , 4 — координата точки, в которой вычислена производная.

Задания для самоконтроля:

1. Найдите угол между касательной в точке $x = 4$ к кривой $y = x^2 + x^5$ и осью Ox .

2. Вычислите производную функции $f(x) = \frac{x-3}{x}$ в точках $x = 3$ и $x = 8$.

Глава 7. Списки в Scilab

В случае, когда требуется хранить большое количество значений какой-либо величины, вместо определения отдельной переменной для каждого значения можно определить список, в котором будет храниться перечень всех значений.

7.1. Создание списков. Обращение к элементам списков

7.1.1. Одномерные списки. Строки и столбцы.

Для создания списка в Scilab необходимо указать его имя и далее, в квадратных скобках, перечислить элементы списка, разделяя их символами «,» или «;», в первом случае элементы списка будут располагаться в строку, во втором — в столбец. Ниже приведены примеры задания списков (Листинг 22).

Листинг 22

<pre>-->spisok1=[1,19,56,100] spisok1 = 1. 19. 56. 100. -- >spisok2=[34;67;78;89] spisok2 = 34. 67. 78. 89.</pre>	<p>Список с именем spisok1 состоит из 4-х элементов. Элементы перечислены через «,», поэтому они расположены в строку, т.е. горизонтально.</p> <p>spisok2 — имя списка; значения списка указаны через «;», поэтому они расположены вертикально, spisoc2 содержит четыре значения.</p>
---	---

Для того чтобы обратиться к нужному элементу списка, достаточно указать имя списка и после него в круглых скобках порядковый номер элемента (Листинги 23 и 24):

<pre>-->spisok1(4) ans = 100.</pre>	<p>Четвертым элементом списка spisok1 является число 100</p>	<p>Листинг 24</p> <pre>-->spisok2(3) ans = 78.</pre>	<p>В данном случае третьим элементом списка «spisok2» является число 78</p>
--	--	---	---

При необходимости можно создать список, элементы которого будут отличаться друг от друга на заданное число. В Scilab создание такого списка выглядит следующим образом:

минимальное значение:шаг:максимальное значение

При этом, если шаг не указан, его значение по умолчанию равно единице.

Рассмотрим примеры задания списков.

- Создадим список, содержащий значения переменной на интервале $[-5; 5]$, с шагом 1.

Решение (Листинг 25):

Листинг 25

```
-->spisok1=-5:5
spisok1 =
-5. -4. -3. -2. -1. 0. 1. 2. 3. 4. 5.
```

В приведенном примере шаг не указан, поэтому он по умолчанию равен единице.

- Создадим список, содержащий значения переменной на интервале $[0; 1,8]$, с шагом 0,2.

Решение (Листинг 26):

Листинг 26

```
-->spisok2=0:0.2:1.8
spisok2 =
0. 0.2 0.4 0.6 0.8 1. 1.2 1.4 1.6 1.8
```

Описанный способ задания списков бывает удобен, когда необходимо вычислить значения некоторой функции или выражения в равноотстоящих точках интервала.

7.1.2. Многомерные списки. Таблицы

Мы рассмотрели формирование одномерного списка (строки или столбца), рассмотрим теперь формирование двумерного списка (таблицы). Такие списки понадобятся нам в дальнейшем, например, для решения систем алгебраических уравнений.

Ввод таблиц осуществляется построчно в следующей последовательности:

1. Указать имя списка, поставить «=», открыть квадратную скобку.
2. Перечислить через запятую элементы первой строки списка, в конце строки запятая не ставится.

3. Нажать клавишу «Enter» и перечислить через запятую элементы следующей строки.

4. Ввести таким же образом нужное количество строк и в конце последней строки закрыть квадратную скобку.

Пример задания таблицы с именем *tablica*, состоящей из 3-х строк и 4-х столбцов (Листинг 27):

Листинг 27

```
-->tablica=[6,45,76,98,0    Выполняем пп. 1-2, Enter;
-->56,78,34,78,67         ввод второй строки, Enter;
-->45,67,89,4,2]         ввод третьей тройки, в конце «]».
tablica =                Ответ Scilab.
  6.   45.  76.  98.  0.
  56  78.  34.  78.
67.
  45.  67.  89.  4.   2.
```

Для обращения к элементу списка необходимо указать имя списка и в круглых скобках через запятую — номер строки и номер столбца, в которых расположен элемент. Обращение к элементу таблицы выглядит следующим образом:

имя_списка(номер_строки,номер_столбца).

Пример обращения к элементу таблицы (Листинг 28):

Листинг 28

```
-->tablica(2,3) 27). Приведенная запись позволяет извлечь указанный элемент таблицы, значение этого элемента может быть присвоено некоторой переменной или использовано в различных вычислениях.
```

В Листинге 29 приведен альтернативный способ задания таблицы. При задании таблицы знак «;» эквивалентен нажатию клавиши «Enter», он также означает переход к следующей строке. Такая запись экономит место в области просмотра.

Листинг 29

```
-->tablica1=[50,67,89,3;4,7,1,0;78,234,45,1]
tablica1 =
  50.  67.  89.   3.
   4.   7.   1.   0.
  78. 234. 45.   1.
```

Примечание. Если задано n строк/столбцов с одинаковым количеством элементов (m), можно объединить их в таблицу. При задании таблицы необходимо указать в квадратных скобках имена строк/столбцов через $;/$, (см. Листинги 30 и 31).

Листинг 30.

```
-->i=[0,1]  Зададим строку из двух элементов и определим переменную  $i$  для ее хранения.
```

```
  i=
  0.  1.
```

```
-->j=[7,2]
```

```
  j=  Зададим строку из двух элементов и определим переменную  $j$  для ее хранения.
```

```
  7.  2.
```

```
--
```

```
>tabl=[i;j]
```

```
  tabl =
```

```
  0.  1.
```

```
  7.  2.
```

В данном случае имеется две строки, содержащих одинаковое число элементов, можно объединить эти строки в таблицу 2×2 . Для создания такой таблицы необходимо задать список, элементами которого будут строки i и j , причем в случае, когда происходит объединение строк, имена строк должны быть перечислены через точку с запятой.

Листинг 31

```
-->i=[0;1]  Зададим столбец из двух элементов и определим переменную  $i$  для его хранения.
```

```
  i=
  0.
```

```
  1.
```

```
-->j=[7;2]
```

```
  j=  Зададим столбец из двух элементов и определим переменную  $j$  для его хранения.
```

```
  7.
```

```
  2.
```

```
-->tabl=[i,j]
```

```
  tabl =
```

```
  0.  7.
```

```
  1.  2.
```

В данном случае имеется два столбца, содержащих одинаковое число элементов, можно объединить эти столбцы в таблицу 2×2 . Для создания такой таблицы необходимо задать список, элементами которого будут столб-

цы i и j , причем в случае, когда происходит объединение столбцов, имена столбцов должны быть перечислены через запятую.

7.2. Математические действия над списком

Если указать имя списка в качестве аргумента математической функции или произвести над списком математическую операцию, то результатом ее действия станет список той же размерности, что и исходный, но его элементами будут результаты действия функции или операции на каждый из элементов исходного списка.

Рассмотрим примеры выполнения математических операций над списками(Листинг 32).

Листинг 32

```

-->spisok1^2
ans =
  1. 361. 3136. 10000.
-->tablica-100
ans =
  - 99.  - 55.  - 24.  - 2.  -
100.
  - 44.  - 22.  - 66.  - 22.  -
33.
  - 55.  - 33.  - 11.  - 96.  -
98.
-->spisok3=[%pi/2,%pi/6,0]
spisok3 =
  1.5707963  0.5235988  0
-->sin(spisok3)
ans =
  1.  0.5  0.
    
```

- Возведем в квадрат все элементы spisok1 (список задан в Листинге 25).
- Вычтем 100 из всех элементов списка tablica (таблица задана в Листинге 27).
- Вычислим $\sin(x)$ в точках $\pi/2, \pi/6, 0$:
 - создадим список spisok3, содержащий значения x ;
 - вычислим значения $\sin(x)$ в этих точках.

Если радиус-векторы на плоскости или в пространстве задавать в виде списков координат, то сложение и вычитание векторов, а также умножение вектора на число можно выполнять, используя операции сложения или вычитания, определенные над списками в Scilab. Например, пусть на плоскости заданы два вектора $\vec{a}=\{2; 7\}$, $\vec{b}=\{4; 8\}$, необходимо найти их сумму и разность (Листинг 33):

```

-->a=[2,7]
a =
  2.  7.
-->b=[4,8]
b =
  4.  8.
-->c1=a+b
c1 =
  6.  15.
-->c2=a-b
c2 =
  -2.  -1.
    
```

Определен список, содержащий координаты вектора a , для удобства будем использовать для его хранения переменную с тем же именем.

Определен список, содержащий координаты вектора b .

Определен список, содержащий координаты вектора-суммы $\vec{c}_1=\vec{a}+\vec{b}$.

Определен список, содержащий координаты вектора-разности $\vec{c}_2=\vec{a}-\vec{b}$.

Задания для самоконтроля:

1. Задайте список-строку, состоящую из элементов 2, 5, 6.
2. Задайте список-столбец, состоящий из тех же элементов.
3. Умножьте на 4 все элементы списка [3,5,1].
4. Задайте две строки из 7 элементов, определив переменные для их хранения, и найдите разности элементов. Объедините строки в таблицу.
5. Задайте две таблицы 4x4 и найдите разность и сумму их элементов. Возведите в квадрат все элементы полученной таблицы. Присвойте переменной t значение элемента, стоящего в первой строке четвертого столбца.

Глава 8. Полиномы

8.1. Задание полиномов. Функция `poly`

Полином n -ой степени имеет вид:

$$a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + \dots + a_nx^n, \quad (1)$$

где a_k — коэффициент при x в степени k ; x — переменная полинома.

Определить полином в Scilab можно при помощи функции **`poly`**.

Есть два способа задания полинома. Главное различие между этими двумя способами состоит в том, что в одном случае полином задается на основе известных коэффициентов полинома, а в другом — на основе известных нулей полинома.

1. Первый способ: полином с заданными коэффициентами.

Пусть необходимо задать полином пятой степени: $k^3 + 8 + 17k^5 + 7k^4$. Для того чтобы задать полином в Scilab, его необходимо представить в виде (1):

$$8 + 0*k + 0*k^2 + 1*k^3 + 7*k^4 + 17*k^5. \quad (2)$$

В нашем примере: $a_0=8, a_1=0, a_2=0, a_3=1, a_4=7, a_5=17$, k — имя переменной полинома.

Вызов функции **`poly`** в данном случае осуществляется следующим образом (Листинг 34):

Листинг 34

```
-->poly([8,0,0,1,7,17],'k','c')
ans =
```

$$8+k^3+7k^4+17k^5$$

- Первым аргументом функции **`poly`** является список, состоящий из коэффициентов полинома, причем первым элементом списка должен быть коэффициент a_0 , вторым — a_1 , третьим — $a_2\dots$, всего $(n+1)$ элементов.
- Вторым аргументом является имя переменной полинома, его следует указывать в одинарных кавычках.
- Третьим аргументом является параметр, который может принимать два значения: s или r . В данном случае его значение — s , это означает что в качестве первого аргумента выступает список коэффициентов полинома.

2. Второй способ: полином с заданными нулями.

Пусть необходимо задать полином, нулями которого являются числа 7 и 2. Вызов функции **`poly`** в данном случае осуществляется следующим образом (Листинг 35):

Листинг 35

```
-->poly([7,2],'m','r')
ans=
```

$$14 - 9f + f^2$$

- Первым аргументом функции `poly` является список, состоящий из корней полинома.
- Вторым аргументом является имя переменной полинома, его следует указывать в одинарных кавычках, в данном случае полином построен для переменной f .
- Третьим аргументом является параметр, который может принимать два значения: s или r . В данном случае его значение — r , это означает, что первый аргумент будет воспринят как список корней полинома, задаваемого функцией **poly**. В результате вызова функции с таким параметром мы получим полином, корнями которого являются элементы списка — первого аргумента функции.

Можно хранить полиномы, задаваемые функцией **poly**, используя переменные. Для этого необходимо указать имя переменной и после знака присваивания «= \Rightarrow » осуществить вызов функции **poly** (см. Листинг 36).

Листинг 36

```
-->j=poly([7,2],'m','r')
j=
```

$$14 - 9f + f^2$$

Для хранения полиномов можно задавать переменные, как это сделано в Листинге 36, эти переменные можно впоследствии использовать при различных операциях с полиномами.

8.2. Действия над полиномами

При необходимости к полиномам могут быть применены математические операции, такие, как сложение полиномов, умножение полиномов, деление полиномов, возведение полинома в степень.

Рассмотрим примеры математических операций над полиномами. Выполним следующие операции:

$$1. \frac{(k^2 + 9k^4)^3}{1 + 9k^2}.$$

Решение (Листинг 37):

Листинг 37

```

--
>p1=poly([0,0,1,0,9],'k','c')
p1 =
    k^2+9k^4
-->p2=poly([1,0,9],'k','c')
p2 =
    1+9k^2
-->l=p1^3/p2
l=
    k^6+18k^8+81k^10
    1

```

Зададим полином k^2+9k^4 , определив переменную $p1$ для его хранения.

Зададим полином $1+9k^2$, определив переменную $p2$ для его хранения.

Возведем полином $p1$ в куб и разделим на полином $p2$, присвоив результат вычисления переменной l .

2. $(h^2-16h^3)*(h^3+h^4+h^2)$.

Решение:

Зададим полиномы h^2-16h^3 и $h^3+h^4+h^2$ и умножим их друг на друга, присвоив результат вычисления переменной $p3$ (Листинг 38).

Листинг 38

```

-->p3=poly([0,0,1,-16],'h','c')*poly([0,0,1,1,1],'h','c')
p3= h^4-15h^5-15h^6-16h^7

```

Такой способ записи удобен, если важен только результат действий над полиномами, а сами полиномы не предполагается использовать в дальнейших вычислениях.

3. $\frac{n^2+4n+4}{n+2}$.

Решение (Листинг 39):

Листинг 39

```

-->poly([4,-4,1],'n','c')/poly([-2,1],'n','c')
ans =
    -2+n
    1

```

Зададим полиномы n^2-4n+4 и $n-2$ и разделим их друг на друга.

1. Задать полином:
 - 1.1. $t+6t^4+t^3+0.5t^2$, определив переменную p для его хранения.
 - 1.2. $5.3k^2-9.8k^3$.
 - 1.3. Задать полином, корнями которого являются числа 6 и -2 .
2. Найти произведение полинома l^3+2l и полинома, корнями которого являются числа 2 и -1 (переменная полинома l).
3. Проверить, являются ли числа -3 и 1 корнями полинома $2x^2+8x+6$.

Глава 9. Решение уравнений

Методику использования Scilab для решения уравнений рассмотрим на примере решения простого уравнения

$$x^3 - 3 = 5. \quad (3)$$

Уравнения, записанные в таком виде, не годятся для Scilab. Прежде всего, необходимо переписать исходное уравнение в виде $f(x)=0$, в таком представлении исходное уравнение примет вид

$$x^3 - 8 = 0. \quad (4)$$

Таким образом, в нашем примере

$$f(x) = x^3 - 8. \quad (5)$$

В общем случае для $f(x)$ должна быть представлена в виде полинома произвольной (целой, положительной) степени переменной x вида

$$f(x) = a_0 + a_1 x^1 + a_2 x^2 + \dots + a_n x^n = \sum_{i=0}^n a_i x^i. \quad (6)$$

Сравнивая последнее выражение с выражением (5), можно видеть, что для получения частного случая $f(x)$ в выражении общего вида следует принять

$$n=3, a_0=-8, a_1=0, a_2=0, a_3=1.$$

Определением значений n и коэффициентов a завершается «математический» этап решения уравнения и можно приступить к формированию задания для Scilab.

На этом этапе, прежде всего, следует сформировать в Scilab полином $f(x)$. Для формирования полиномов используется функция **poly**. Так, для описания $f(x)$ из нашего примера обращение к функции **poly** будет выглядеть следующим образом:

$$\mathbf{g} = \mathbf{poly}([-8, 0, 0, 1], 'x', 'c').$$

Здесь результат выполнения функции **poly** присваивается переменной с именем g . Из приведенного выражения видно, что функция **poly** имеет три параметра. Первый параметр — это список коэффициентов полинома. Следуя правилам, изложенным в п. 7.1, коэффициенты полинома перечислены через запятую и заключены в квадратные скобки. Следующий параметр — ' x ' — задает имя переменной, для которой записан полином. И наконец, ' c ' означает, что первый параметр — это коэффициенты полинома.

Третий параметр функции **poly** может принимать два значения ' c ' и ' r '. Если ' c ' заменить на ' r ' при обращении к функции, то при формировании полинома Scilab будет рассматривать первый параметр как список корней полинома, а не его коэффициенты. В результате

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab
 функция **poly** построит полином, имеющий корни из списка, заданного в первом аргументе. Этот способ может быть удобен при формировании вариантов заданий, например для контрольных и самостоятельных работ по математике.

Последний этап вычисления корней уравнения состоит в вызове встроенной функции Scilab с именем **roots** (корни, англ.):

res=roots(g).

Функция **roots** имеет один обязательный аргумент, который представляет собой полином, созданный функцией **poly**. В данном случае результат выполнения функции или, иначе говоря, возвращаемое значение помещается в переменную с именем *res*. В листинге 40 приведено решение исходного уравнения в Scilab.

Листинг 40

--	Формируем полином. Нажимаем Enter.
>g=poly([-8,0,0,1],'x','c')	Сравниваем получившийся полином с
g =	формулой, проверяем правильность.
-8+x ³	Ищем корни уравнения .
-->res=roots(g)	
res =	В результате имеем список из трех кор-
2.	ней, один вещественный, два мнимых.
-1. +1.7320508i	
-1. -1.7320508i	

Из листинга видно, что кроме очевидного корня $x=2$ функция **roots** возвращает два комплексно-сопряженных корня $-1+1.7320508i$ и $-1-1.7320508i$ ($i=\sqrt{-1}$ — мнимая единица). И поэтому переменная *res*, которой присваивается результат, будет содержать список из трех значений. Отметим, что в курсе средней школы комплексная арифметика не изучается, поэтому объяснение результата учащимся может представлять определенную методическую трудность.

Если найденный вещественный корень уравнения необходимо использовать в дальнейших вычислениях, то можно выделить его из списка, используя правила работы со списками, описанными в п. 7.1. Из листинга видно, что вещественный корень является первым элементом в списка корней уравнения, содержащегося в переменной *res*. Извлечение первого корня из списка решений и присваивание его значения переменной, которую назовем *x*, выполнится следующим образом: $x=res(1)$. В результате выполнения этой строки в командной строке Scilab переменная *x* получит значение 2.

Рассмотрим пример решения квадратного уравнения

$$x^2 - x - 2 = 0 . \quad (7)$$

Как и ранее, сначала формируем список коэффициентов при степенях переменной x в формуле (7), для рассматриваемого уравнения список будет состоять из трех элементов (8).

$$a_0 = -2, a_1 = -1, a_2 = 1 . \quad (8)$$

После этого можно приступать к формулировке задания для Scilab. В соответствии с алгоритмом решения, рассмотренным выше, теперь необходимо обратиться к функции **poly** для формирования полинома, после чего использовать сформированный полином в качестве аргумента функции **roots**, которая вычислит корни уравнения. Реализация этой последовательности действий в предыдущем примере заняла две строки. Ниже приведен более короткий вариант реализации того же алгоритма (9), но теперь уже применительно к решению квадратного уравнения:

$$\mathbf{roots (poly ([-2,-1,1], 'x', 'c'))} . \quad (9)$$

Отметим, что при решении квадратных уравнений **roots** всегда будет выдавать два корня. В зависимости от дискриминанта D эти корни будут вещественными ($D \geq 0$) или же комплексными (при $D < 0$). Так что результат вычислений может противоречить утверждению об отсутствии корней квадратного уравнения при $D < 0$, принятому в школьном курсе математики.

Выражение типа (9) может использоваться как шаблон для решения уравнения любого порядка. Так, например, для решения биквадратного уравнения

$$x^4 - 13x^2 + 12 = 0 . \quad (10)$$

Задание для Scilab выглядит аналогично:

$$\mathbf{roots(poly([12, 0, -13, 0, 1], 'x', 'c'))} .$$

разница между этим выражением и выражением (9) состоит лишь в количестве и значениях коэффициентов a_n при степенях x , входящих в формулу. В общем виде задание для Scilab для решения уравнения степени n можно записать в виде

$$\mathbf{roots(poly([a0,a1,a2,...,an], 'x', 'c'))} . \quad (11)$$

Листинг программы для Scilab, где уравнения (7) и (10) решены с помощью выражения общего вида (11), приведен в Листинге 41.

Листинг 41

<pre>-->roots(poly([-8,0,0,5],'x','c')) ans = 1.1696071 -0.5848035+1.0129095i -0.5848035-1.0129095i -->roots(poly([-2,-1,1],'x','c')) ans = - 1 2 -->roots(poly([12,0,-13,0,1.],'x','c')) ans = 1. -1. -3.4641016 3.4641016</pre>	<p>Решение кубического уравнения (3).</p> <p>Решение квадратного уравнения (7).</p> <p>Решение биквадратного уравнения (10).</p>
---	--

Таким образом, численное решение уравнений вида $\sum_{i=0}^n a_i x^i = 0$ может быть получено в Scilab с использованием выражения (11). При этом следует иметь в виду, что в результате численного решения уравнений, наряду с вещественными корнями, могут получаться комплексные корни, что, вероятно, вызовет некоторые методические неудобства, поскольку комплексные числа не изучаются в школьном курсе математики.

Задания для самоконтроля:

Решите уравнения:

1. $x^2 + 2x - 15 = 0$.
2. $7x^2 + 5x = 0$.
3. $5b^4 - 15b^3 + 6 = 0$.

Глава 10. Решение систем линейных уравнений

Рассмотрим решение системы из трех уравнений с тремя неизвестными переменными x, y, z :

$$\begin{cases} 3x - 6y + 9z = 16, \\ z - 15x = 0, \\ 6y - 7 - 8x + 2z = 3. \end{cases}$$

Прежде чем приступить к формированию задачи для Scilab, необходимо привести систему уравнений к виду

$$\begin{cases} a_1x + b_1y + c_1z = k_1, \\ a_2x + b_2y + c_2z = k_2, \\ a_3x + b_3y + c_3z = k_3. \end{cases}$$

Для нашего примера система будет выглядеть следующим образом:

$$\begin{cases} 3x - 6y + 9z = 16, \\ -15x + 0y + 1z = 0, \\ -8x + 6y + 2z = 10. \end{cases}$$

Для решения подобных систем уравнений в Scilab существует функция **linsolve**. Обращение к ней выглядит следующим образом:

linsolve(K,k).

- K — таблица, составленная из коэффициентов уравнений системы (см. пп 7.1.2), причем она сформирована таким образом, что каждая строка представляет собой список коэффициентов одного из уравнений системы, а каждый столбец — список коэффициентов при одноименных переменных, то есть если первым элементом в первой строке является коэффициент при y , то первыми элементами других строк также должны быть коэффициенты при y в соответствующих уравнениях.

Общий вид K :

$$K = \begin{matrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{matrix}.$$

Для решаемой системы:

$$K = \begin{matrix} 3 & -6 & 9 \\ -15 & 0 & 1 \\ -8 & 6 & 2 \end{matrix}.$$

- k — столбец, содержащий свободные (стоящие после знака «=») коэффициенты.

Примечание: при задании в Scilab k должен быть именно **столбцом**, поэтому перечисление переменных нужно делать через «;» (см. пп. 7.1.1).

Общий вид k :

$$k = \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix}.$$

Для решаемой системы

$$k = \begin{pmatrix} 16 \\ 0 \\ 10 \end{pmatrix}.$$

После того как элементы списков K и k определены, приступим к решению системы в Scilab (Листинг 42).

Листинг 42

```
-->K=[3,-6,9
-->-15,0,1
-->-8,6,2]
```

```
K =
  3.  -6.  9.
-15.  0.  1.
 -8.  6.  2.
```

```
-->k=[16;0;10]
```

```
k =
 16.
  0.
 10.
```

```
-->reshenie=linsolve(K,k)
reshenie =
-0.1625
-1.0708333
-2.4375
```

Зададим таблицу коэффициентов при неизвестных переменных.

Зададим столбец независимых коэффициентов.

Решим уравнение при помощи функции `linsolve` (результат решения в данном случае присваивается переменной «reshenie»).

Результатом решения станет список значений переменных x, y, z , причем значения переменных расположены в столбце в том порядке, в котором были расположены столбцы с коэффициентами при этих переменных в таблице K .

Задания для самоконтроля:

Решить системы уравнений:

$$\begin{cases} 5x - 8y = 0, \\ x - \frac{1}{6}y = 1. \end{cases}$$

$$\begin{cases} 3x + 15y = 0, \\ y + 16x. \end{cases}$$

Глава 11. Построение графиков

Scilab содержит набор функций для графического представления информации. Для построения графиков функции одной переменной, например, служат функции **plot** и **polarplot**. Первая — позволяет строить графики в декартовых координатах, вторая — изображает графики в полярных координатах. Обе эти функции требуют предварительного определения списка значений аргумента и списка соответствующих значений функции, график которой необходимо построить. Списки аргументов и значений функции указываются первым и вторым аргументами при вызове **plot** и **polarplot**. Ниже подробнее будет рассмотрено использование функции **plot**. Она же может использоваться для построения графиков в полярных координатах, используя очевидную связь между декартовыми и полярными координатами.

11.1. Функция **plot**

Функция **plot** предназначена для построения двумерных графиков функции одной переменной вида $f=f(x)$.

Обращение к функции **plot** выглядит следующим образом:

plot(x,y),

где x — список значений независимой переменной, а y — список значений функции f в этих точках. Соответственно для того, чтобы построить график в Scilab, необходимо создать список значений переменной и список значений функции в этих точках.

График, построенный при помощи функции **plot**, представляет собой ломаную линию, соединяющую значения функции, вычисленные в заданных точках. Следовательно, чтобы получить гладкий график, необходимо вычислить значение функции в большом количестве точек.

Рассмотрим пример построения графика функции $\sin^2(2x)$ (Листинг 43).

Листинг 43

```
-->x=-
%pi:0.01:%pi;
```

```
-->y=(sin(2*x))^2;
```

```
-->plot(x,y)
```

Список x получает значения от $-\pi$ до π с шагом 0.01.

Список y содержит значения функции для каждого элемента в x .

Строим график.

В результате обращения к функции **plot** открылось окно графического редактора Scilab, содержащее график функции (рис. 3).

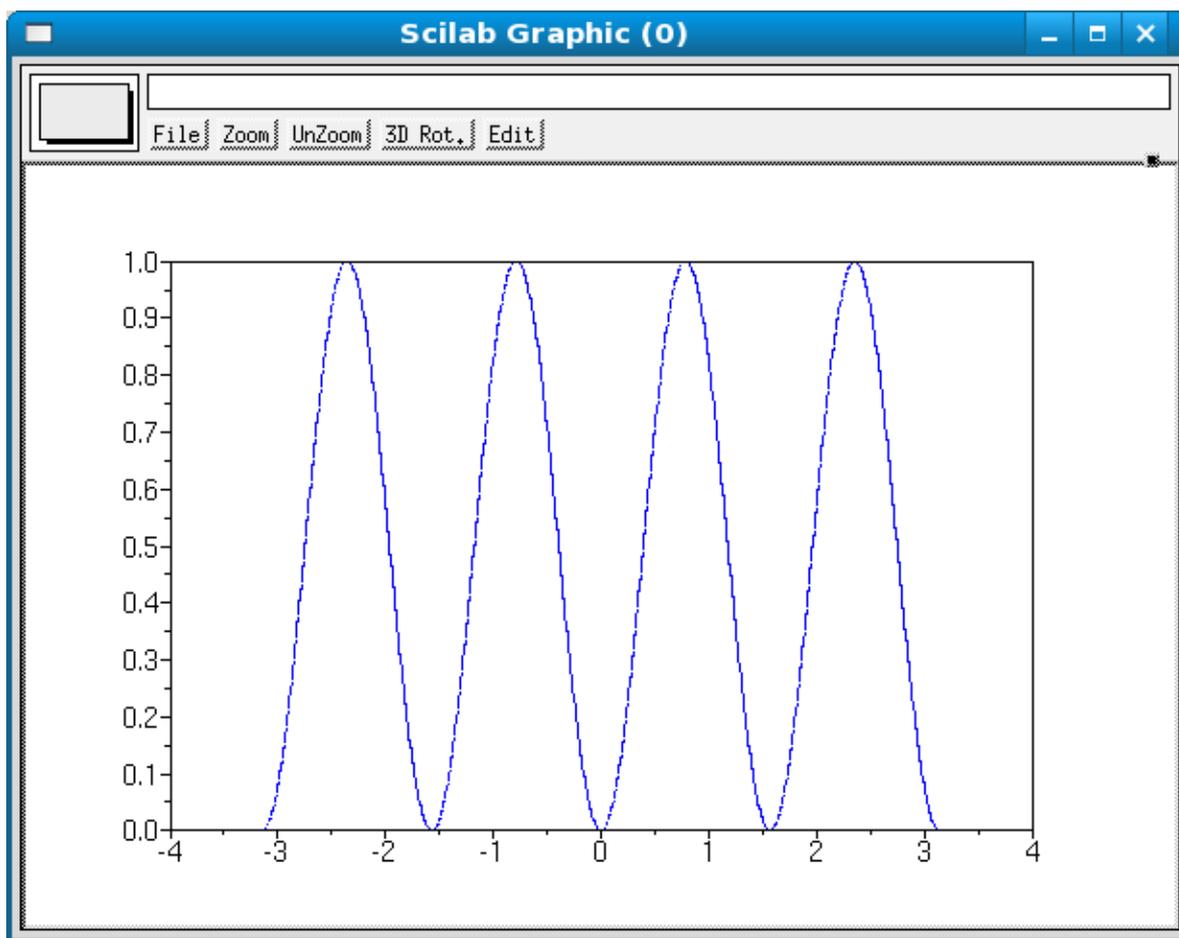


Рис. 3. Графическое окно Scilab

Из-за того, что значение функции было вычислено в большом количестве точек, т.е. был выбран малый шаг разбиения промежутка построения, равный 0.01, график функции получился гладким без видимых изломов.

При последующем обращении к функции **plot** в текущем сеансе график будет построен в том же окне и линия будет того же цвета, что и у предыдущего графика. Для построения нескольких графиков в одной области удобнее использовать обращение к функции `plot` следующего вида: **plot(x1,y1,x2,y2,x3,y3)**, где $x1$ — список значений переменной x , в которых должно быть вычислено значение функции $f1=y1(x)$, $x2$ — список значений переменной x , в которых должно быть вычислено значение функции $f2=y2(x)$, и т.д. Если построение происходит в одной области, в качестве $x1$, $x2$, $x3$ может выступать один и тот же список.

Построим в одной области графики функций: $f1=\sin(x)$; $f2=\cos(x)$; $f3=\sqrt{|x|}$ (Листинг 44).

Листинг 44

```
-->x=-2*%pi:0.01:2*%pi;  
-->f1=sin(x);  
-->f2=cos(x);  
-->f3=sqrt(abs(x));  
-->plot(x,f1,x,f2,x,f3)
```

В данном случае список x является списком аргументов для всех трех функций.

Результат работы функции **plot**, обращение к которой содержит Листинг 44, можно видеть на рис. 4.

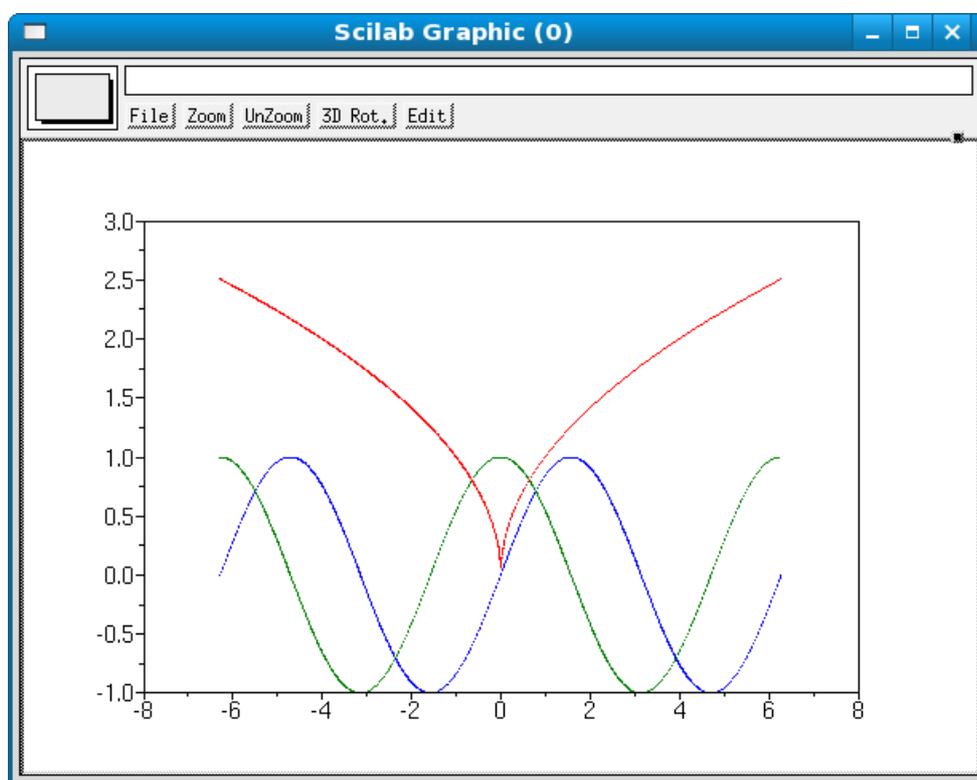


Рис. 4. Графическое окно Scilab

Как видно, графики функций в данном случае автоматически изображаются разными цветами.

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

Строка `string` выглядит следующим образом

'параметр1параметр2параметр3'.

Символы пишутся один за другим без разделителей.

- Параметр1 определяет цвет линии графика:

Таблица 11.1

Значения параметра функции plot, определяющего цвет графика

Символ	Описание
y	желтый
m	розовый
c	голубой
r	красный
g	зеленый
b	синий
w	белый
k	черный

- Параметр2 определяет тип маркера графика:

Таблица 11.2

Значение параметра, определяющего тип маркеров (точек) графика

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

- Параметр 3 определяет тип линии графика:

Таблица 11.3

Значения параметра, определяющего тип линии графика

Символ	Описание
-	сплошная
:	пунктирная
-.	штрихпунктирная
--	штриховая

Если один из символов не указан, его значение выбирается по умолчанию, при неуказанном типе маркера — маркер будет отсутствовать.

Пример:

Построим 3 графика на отрезке $x \in [0; 3]$:

$f1(x)=x^2$ — пунктирная линия желтого цвета;

$f2(x)=1+x$ —штриховая линия голубого цвета;

$f3(x)=e^x$ —сплошная линия черного цвета.

Решение (Листинг 45):

Листинг 45

```
-->x=0:0.1:3;
-->f1=x^2;
-->f2=x+1;
-->f3=%e^x;
--
>plot(x,f1,'y:',x,f2,'c--',x,f3,'k-')
```

Зададим список точек с именем x , в которых будут вычислены значения функций.

Вычислим значения функции $f1$ в точках списка x .

Вычислим значения функции $f2$ в точках списка x .

Вычислим значения функции $f3$ в точках списка x .

Построим по вычисленным значениям графики функций с заданными параметрами (см. табл. 11.1 и 11.3).

В результате выполнения команд, приведенных в Листинге 45, в графическом окне Scilab будут построены графики функций $f3(x)$, $f2(x)$, $f1(x)$ (рис. 5).

11.2. Оформление графиков

11.2.1. Изображение сетки в графической области

Для изображения сетки следует воспользоваться функцией **xgrid(color)**. Color — аргумент функции, он определяет цвет линий сетки. Общий вид аргумента: $a_1a_2a_3$, где каждый из параметров a_1 , a_2 и a_3 может принимать два значения — 0 или 1. Параметр a_1 определяет наличие (при $a_1=1$) или отсутствие (при $a_1=0$) красной составляющей в цвете линии, a_2 — зеленой и a_3 — синей. Различные комбинации значений этих параметров позволяют получать различные цвета. Например, в результате вызова функции **xgrid(100)** в графической области будет отображена сетка красного цвета. Значения параметров следует вводить подряд без разделителей.

11.2.2. Вывод названий графика и осей

Вывод названий осуществляется с помощью функции **xtitle(name, xlabel, ylabel)**, здесь name — название графика, xlabel — название оси X, ylabel — название оси Y.

11.2.3. Нанесение описания линий

Нанесение осуществляется вызовом функции `legend`. Обращение к этой функции выглядит следующим образом:

`legend(line1,line2,...,linen,place,frame)`

- `line1` — описание (название) первого графика, `line2` — второго ..., `linen` — имя n -го графика;
- `place` определяет месторасположение описания: если `place=1`, то описание будет расположено в верхнем правом углу графической области (если значение `place` не указано, то по умолчанию `place=1`), 2 — в верхнем левом углу, 3 — в нижнем левом углу, 4 — в нижнем правом углу, 5 — положение определяется пользователем после изображения графика;
- `frame` — может принимать два значения `%t` и `%f` (по умолчанию `frame=%t`), если `frame=%t`, то описание будет заключено в рамку, если же `frame=%f` — рамка отсутствует.

Задания для самоконтроля:

1. Построить в одном графическом окне графики функций x^2 и x^5 на промежутке от -5 до 5 . Нанести сетку на график.
2. Построить графики $\sin(x)$ и $\cos(x)$ на промежутке от $-\frac{\pi}{2}$ до $\frac{\pi}{2}$ и нанести подписи линий.
3. Построить три графика произвольных функций одного цвета в одном графическом окне.

Глава 12. Примеры вычислений в Scilab

12.1. Вычисление площади круга и длины окружности

Вычислить площадь круга и длину окружности радиуса $r=5.78$.

Решение:

Площадь круга и длина окружности вычисляются по формулам: $S=\pi*r^2$, $l=2*\pi*r$ соответственно, где r — радиус круга/окружности. Вычислим площадь круга и длину окружности радиуса $r=5.78$ (см гл. 3). В Листинге 46 показано вычисление площади круга и длины окружности при помощи Scilab.

Листинг 46

```
-->S=%pi*5.78^2
S =
    104.95558
-->l=2*%pi*5.78
l =
    36.316811
```

12.2. Вычисление длины вектора

Пусть $\vec{a}=\{a_1, a_2\}$ — вектор с координатами a_1 и a_2 . Длина вектора вычисляется по формуле $|\vec{a}|=\sqrt{a_1^2+a_2^2}$. Вычислим длину вектора $\vec{a}=\{3.67, 5.78\}$ по формуле $|\vec{a}|=\sqrt{3.67^2+5.78^2}$ (см. гл. 3). Команды для вычисления длины вектора с помощью Scilab приведены в Листинге 47.

```
-->a1=3.67
a1 =
    3.67
-->a2=5.78
a2 =
    5.78
-->a=sqrt(a1^2+a2^2)
a =
    6.8466999
```

12.3. Построение касательной к кривой в заданной точке

Построить график функции $f(x)=5-x^2$ и касательной к графику функции в точке $x=2$.

Решение:

Касательная прямая определяется уравнением $l(x)=kx+b$, где $k=f'(x_0)$, $b=f(x_0)-k*x_0$, $x_0=a$ — координата точки касания.

Решение задачи в Scilab (Листинги 48 — 52):

1. Зададим функцию $f(x)=5-x^2$, и значение $x_0=2$ координаты точки касания (см. п. 2.1 и гл. 5) (Листинг 48).

Листинг 48

```
-->function d=f(x), d=5-x^2 endfunction
-->x0=2;
```

Определим функцию $f(x)=5-x^2$.

Определим x_0 — координату точки касания.

2. Для того, чтобы построить касательную прямую необходимо сначала определить коэффициенты b и k в уравнении прямой (Листинг 49).

Листинг 49

```
-->k=numdiff(f,x0)
k =
-4.
```

Определим коэффициент k . Для этого с помощью функции **numdiff** (см гл. 6) найдем производную функции в точке $x_0=2$.

```
-->b=f(x0)-k*x0
b =
9.
```

Определим коэффициент b по формуле: $b=f(a)-k*x_0$ (см. гл. 3). Строка содержит вызов определенной нами функции с именем «f».

3. Определим функцию $kas(x)=k*x+b$ (см гл. 5) (Листинг 50), график этой функции является касательной прямой к графику функции $f(x)=5-x^2$ в точке $x_0=2$.

Листинг 50

```
-->function g=kas(x), g=k*x+b endfunction
```

4. Определим $xgraph$ — список точек для построения графика (см. пп. 7.1.1) (Листинг 51).

Листинг 51

```
--  
>xgraph=-5:0.01:5;
```

Определим список, содержащий значения x на интервале от -5 до 5 с шагом $0,01$.

5. Вычислим значения функций $f(x)$ и $kas(x)$ в точках из списка $xgraph$ (см п 7.2) (Листинг 52).

Листинг 52

```
-->fgraph=f(xgraph);  
-->kasgraph=kas(xgraph);
```

Аргументом функций $f(x)$ и $kas(x)$ является список $xgraph$, это значит что значение функции будет вычислено во всех точках списка. Результатом вызова функции $kas(xgraph)$ будет список значений функции в точках списка $xgraph$.

6. Используя функцию **plot** (см. п. 11.1), построим графики функций $f(x)$ и $kas(x)$ (Листинг 53).

Листинг 53

```
-->plot(xgraph,fgraph,xgraph,kasgraph)
```

Используя списки $xgraph$, $fgraph$, $kasgraph$, изобразим графики функций $f(x)$ и $kas(x)$ (рис. 6).

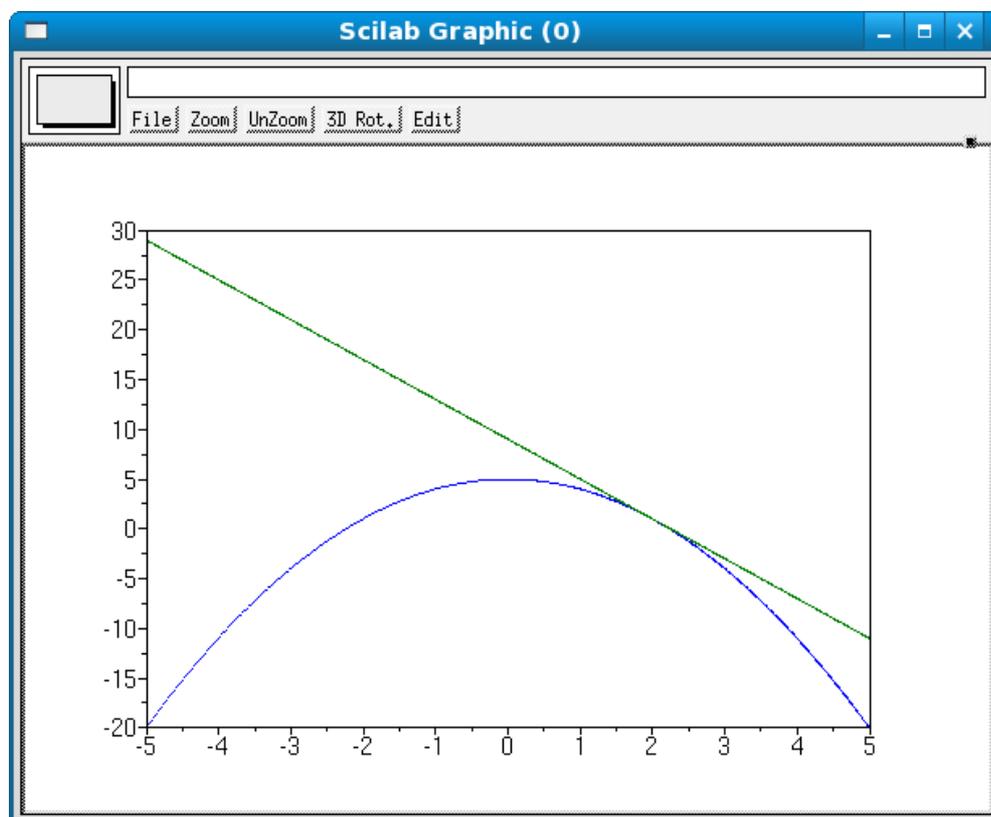


Рис. 6. Результат вызова функции plot

12.4. Решение уравнения графическим способом

Пусть необходимо решить уравнение $e^x - x^2 = 0$.

Решение:

Преобразуем исходное уравнение к виду $y_1(x) = y_2(x)$:

$$e^x = x^2 .$$

Чтобы найти корни уравнения, необходимо найти точку пересечения графиков функций $y_1(x) = e^x$ и $y_2(x) = x^2$. Построим графики функций (см. п. 11.1) в Scilab и нанесем на график сетку для удобства визуального определения точки пересечения (см. пп. 11.2.1). Необходимые команды приведены в Листинге 54, результат выполнения команд можно видеть на рис. 7.

Листинг 54

```
-->x=-2:0.01:2;  
-->y1=%e^x;  
-->y2=x^2;  
-->plot(x,y1,x,y2)  
-->xgrid(001)
```

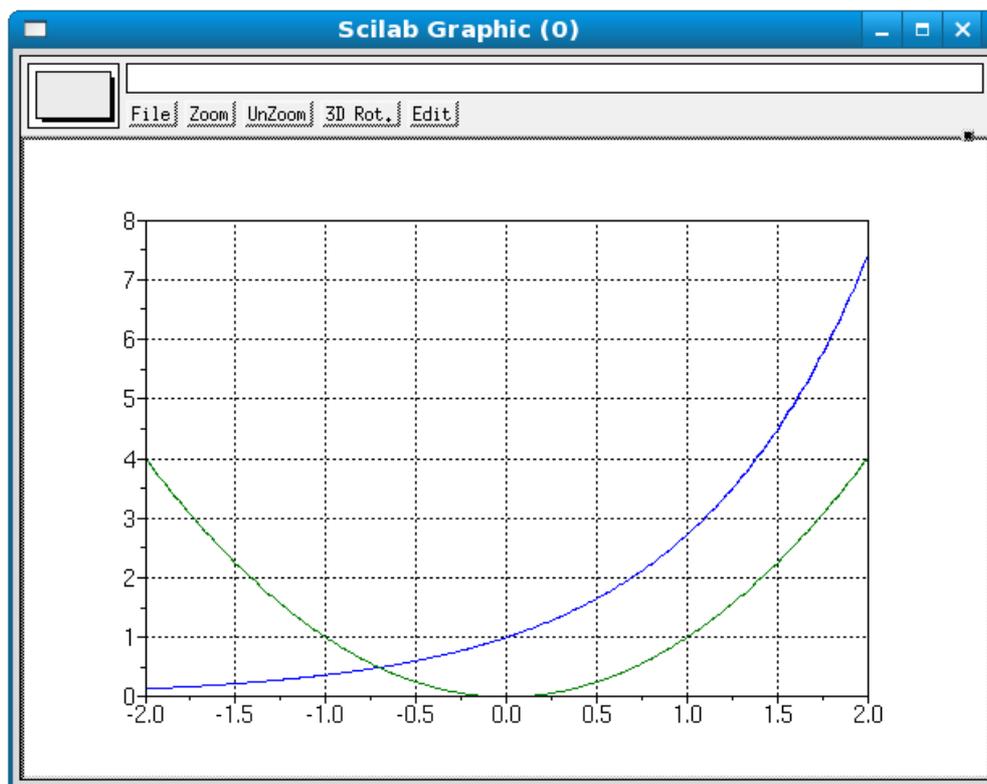


Рис. 7. Графики функций

Графики пересекаются при $x \approx -0.7$ — это и есть искомым корень уравнения.

12.5. Решение биквадратного уравнения

Найти корни уравнения $5x^4 - x - 18 = -2x^2$.

Решение:

Приведем уравнение к виду $f(x) = 0$:

$$5x^4 + 2x^2 - x - 18 = 0.$$

Поиск нулей полинома при помощи Scilab (см. гл. 9, выражение (11) приведен в Листинге 55.

Листинг 55

```
-->roots(poly([-18,-1,2,0,5],'x','c'))
ans =
  1.3328559
 -1.2804456
 -0.0262051 +1.4521397i
 -0.0262951 -1.4521397i
```

12.6. Вывод значения выражения на экран

Вывести на экран значение числа $\frac{\sqrt{3}\pi}{9}$ до двенадцатого знака после запятой.

Решение:

Для вывода числа на экран необходимо использовать функцию `printf` (см. п. 4.2) (Листинг 56):

Листинг 56

```
-->printf("%.12f",sqrt(3)*%pi/9)  Выводимое значение вычислено во
0.0604599788078                время вызова функции printf.
```

12.7. Вычисление определенного интеграла

Пусть необходимо вычислить определенный интеграл $\int_0^{3\pi} \frac{1}{\cos^2 \frac{x}{9}} dx$.

Решение:

Определим функцию с именем s , которая будет задавать выраже-

ние $\frac{1}{\cos^2 \frac{x}{9}}$ (см. гл. 5) и вычислим интеграл по заданным пределам при помощи функции **numdiff** (см. п. 6.2). Листинг 57 демонстрирует определение функции и вычисление определенного интеграла по заданным пределам от этой функции.

Листинг 57

```
-->function c=s(x), c=1((cos(x/9))^2) endfunction
-->intg(0,3*%pi,s)
ans =
    15.588457
```

12.8. Вычисление длины гипотенузы

Пусть требуется найти гипотенузу прямоугольного треугольника, если длины его катетов равны 5 и 3.

Решение:

Длина гипотенузы (c) вычисляется по формуле $c = \sqrt{a^2 + b^2}$, где a и b — длины катетов треугольника. Воспользуемся математическими операциями для вычислений (см. гл. 3). Вычисление длины гипотенузы по заданным длинам катетов при помощи Scilab приведено в Листинге 58.

Листинг 58

```
-->a=5;
-->b=3;
-->c=sqrt(a^2+b^2)
c =
    5.8309519
```

12.9. Задание функции для вычисления логарифма с произвольным основанием

Согласно свойствам логарифма:

$$\log_a b = \frac{\log_2 b}{\log_2 a}.$$

$\log_2 a$ является встроенной функцией Scilab, следовательно, логарифм с любым основанием можно вычислить при помощи встроенной функции $\log_2(x)$ (см. п. 4.1).

Для вычисления зададим функцию (см. гл. 5), аргументами которой будут основание логарифма и число, логарифм которого необходимо

И.С. Тропин, О.И. Михайлова, А.В. Михайлов Численные и технические расчеты в среде Scilab
вычислить. Функция будет вычислять логарифм числа по произвольному основанию по формуле: $\log_a b = \frac{\log_2 b}{\log_2 a}$. Реализация таких вычислений в Scilab показана в Листинге 59.

Листинг 59

```
-->deff('k=LOG(osn,arg)','k=(log2(arg))/(log2(osn))');  
-->LOG(3,9)  
ans =  
    2.
```

В данном примере *LOG* — имя функции, *osn* и *arg* — входные параметры функции, они задают основание логарифма и число, которое необходимо прологарифмировать, соответственно.

Во второй строке Листинга 59 проведено обращение к функции *LOG*. Такое обращение должно привести к вычислению логарифма по основанию 3 от 9 (3 — основание логарифма, 9 — аргумент логарифма).

Задания для самоконтроля

1. Материальная точка движется прямолинейно по закону

$$x(t) = -\frac{1}{3}t^3 + t^2 + 5t.$$

2. Найти скорость точки в момент $t=2$ с (перемещение измеряется в метрах).

3. Найдите силу, действующую на материальную точку массой $m=5$ кг, движущуюся прямолинейно с переменной скоростью $v=6t^2-2t$ при $t=2$ с.

4. Вращение тела вокруг оси совершается по закону $\varphi(t)=3t^2-4t+2$, найдите угловую скорость при $t=4$ с.

5. Материальная точка движется прямолинейно по закону

$$x(t) = \frac{\sqrt{5t+1}}{3t^3-8t+2}.$$

Найти скорость точки в момент $t=5$ с. (перемещение измеряется в метрах).

6. Вычислить определенный интеграл $\int_1^4 (x + \frac{\sqrt{x}}{x}) dx$.

7. Сила в 4 Н растягивает пружину на 8 см. Какую работу надо произвести, чтобы растянуть пружину на 8 см?

8. Вычислить определенный интеграл $\int_0^{3\pi} \frac{dx}{\sqrt[5]{\cos^2 \frac{x}{9}}}$.

9. Определите точку пересечения касательной в точке $x=3$ к графику функции $f(x)=8x-2x^2$ с осью Ox .

10. Постройте график функции $\cos \frac{\sqrt{x^2+3x+5}}{2}$ и определите по нему

точки минимума и максимума функции на промежутке $[\frac{\pi}{2}; 6\pi]$.

11. Вычислить скалярное произведение векторов, если $|\vec{a}|=2$, $|\vec{b}|=3$, а угол между ними равен $\frac{\pi}{3}; \frac{\pi}{6}; \frac{\pi}{2}$.

12. Докажите, что векторы $\vec{i}+\vec{j}$ и $\vec{i}-\vec{j}$ перпендикулярны, если \vec{i} и \vec{j} — координатные векторы.

13. Решить систему уравнений:

$$\begin{cases} 5x-8y=0, \\ x-\frac{1}{6}y=1. \end{cases}$$

14. Задать функцию для вычисления длины вектора по его координатам.

15. Найдите координаты вектора \vec{v} , если $\vec{v}=2\vec{a}-3\vec{b}-\frac{1}{2}\vec{c}$, $\vec{a}=\{4; 1\}$,
 $\vec{b}=\{1; 2\}$, $\vec{c}=\{4; -6\}$.

Глоссарий

В

Внешняя функция — функция, определенная для использования в различных сеансах Scilab.

Встроенная функция — функция, описание которой хранится в специальном файле (библиотеке). Этот файл подгружается автоматически при запуске Scilab.

Г

Графический интерфейс пользователя (ГИП, англ. graphical user interface, GUI) в вычислительной технике — система средств для взаимодействия пользователя с компьютером, основанная на представлении всех доступных пользователю системных объектов и функций в виде графических компонентов экрана (окон, значков, меню, кнопок, списков и т. п.). При этом, в отличие от интерфейса командной строки, пользователь имеет произвольный доступ (с помощью клавиатуры или устройства координатного ввода типа "мышь") ко всем видимым экран-ным объектам.

К

Комментарии — пояснения к исходному тексту программы, оформленные по правилам, определяемым языком программирования. Комментарии не оказывают никакого влияния на результат работы программы.

Л

Листинг — текст компьютерной программы или ее части, записанный на каком-либо языке программирования. Обычно данный термин употребляется для текста, распечатанного на бумаге.

О

Оператор (в программировании) — фраза алгоритмического языка, определяющая законченный этап обработки данных. В состав операторов входят ключевые слова, данные, выражения и др.

Различают:

атомарные операторы, никакая часть которых не является самостоятельным оператором;

составные операторы, объединяющие другие операторы в новый, укрупненный оператор.

С

Список — упорядоченная совокупность объектов, называемых элементами списка.

Т

Терминал — оконечная часть некой системы, которая обеспечивает связь системы с внешней средой. Терминал в Unix представляет собой системное приложение.

Ф

Функция (в языке программирования) — это заданная последовательность действий, реализующая определенный алгоритм. Определяется описанием типа результата, именем, списком формальных параметров и составным оператором, описывающим выполняемые функцией действия.

Ч

Численное дифференцирование — это вычислительная процедура определения производной в заданной точке. Простейшая идея численного дифференцирования состоит в том, что функция заменяется интерполяционным многочленом и производная функции приближенного заменяется соответствующей производной интерполяционного многочлена.

Численное интегрирование (историческое название: квадратура) — вычисление значения определенного интеграла (как правило, приближенное), основанное на том, что величина интеграла численно равна площади криволинейной трапеции, ограниченной осью абсцисс, графиком интегрируемой функции и отрезками прямых $x=a$ и $x=b$, где a и b — пределы интегрирования.

Список литературы

Рекомендуемая

1. Андреевский Б., Фрадков А. Элементы математического моделирования в программных средах MatLab 5 и Scilab. — СПб.: Наука, 2001. — 286 с.
2. Алексеев Е.Р., Чеснокова О.В. Scilab — теория и практика на русском языке / Е.Р. Алексеев, О.В. Чеснокова — 2007. — Режим доступа: <http://www.scilab.land.ru>
3. Павлова М.И. Руководство по работе с пакетом Scilab / М.И. Павлова. — Режим доступа: http://www.csa.ru/~zebra/my_scilab/

Использованная

1. Campbell S.L., Nikoukhah R. Auxiliary Signal Design for Failure Detection. — Cloth, 2004. — 208 p.
2. Bunks C., Chancelier J.P., Delebecque F., et al. [Engineering and Scientific Computing with Scilab](#) — Boston.: Birkhäuser, 1999. — 524 p.
3. Campbell S.L., Chancelier J.P., Nikoukhah R. Modeling and Simulation in Scilab/Scicos – Springer, 2005. — 313 p.

Приложение А

Оформление текста программы для определения внешней функции.

```
function[имя1,...,имяM]=имя_функции(переменная_1,...,переменная_N)  
тело функции  
endfunction
```

В данном случае **имя1,...,имяM** – список выходных параметров (от 1 до M), то есть переменных, которым будет присвоен конечный результат вычислений, **имя_функции** – имя, с которым эта функция будет вызываться, **переменная_1,...,переменная_N** – входные параметры (N параметров).