

**А. М. Половко**

# **Derive**

**ДЛЯ СТУДЕНТА**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06(075.8)  
ББК 32.973.26-018.2я73  
П52

**Половко А. М.**

П52 Derive для студента. — СПб.: БХВ-Петербург, 2005. — 352 с.: ил.

ISBN 5-94157-594-7

Содержится краткое описание методов решения математических задач и подробные технологии их реализации с помощью системы компьютерной алгебры Derive на примере версии 5. Описаны элементы программирования на языке системы. Приведены примеры программ вычисления функций, решения уравнений, вычисления интегралов. Представлены задачи повышенной сложности с учетом интеллектуальных возможностей системы.

*Для студентов, аспирантов, преподавателей технических вузов  
и специалистов, применяющих математические вычисления  
в профессиональной деятельности*

УДК 681.3.06(075.8)  
ББК 32.973.26-018.2я73

#### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Людмила Еремеевская</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Владимир Красильников</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Игоря Цырульников</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 03.02.05.

Формат 60×90<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 22.

Тираж 3000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию  
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой  
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов

в ГУП "Типография "Наука"

199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-594-7

© Половко А. М., 2005

© Оформление, издательство "БХВ-Петербург", 2005

# Оглавление

<b>Введение .....</b>	<b>9</b>
Особенности книги .....	11
Для кого эта книга .....	12
 <b>Глава 1. Основы работы с системой Derive 5 .....</b>	<b>13</b>
1.1. Derive 5 как универсальная система символьной математики .....	13
1.2. Главное окно системы .....	16
1.3. Главное меню системы .....	22
1.3.1. Пункт меню <i>File</i> .....	22
1.3.2. Пункт меню <i>Edit</i> .....	28
1.3.3. Пункт меню <i>Insert</i> .....	29
1.3.4. Пункт меню <i>Author</i> .....	30
1.3.5. Пункт меню <i>Simplify</i> .....	32
1.3.6. Пункт меню <i>Solve</i> .....	35
1.3.7. Пункт меню <i>Calculus</i> .....	36
1.3.8. Пункт меню <i>Declare</i> .....	36
1.3.9. Пункт меню <i>Options</i> .....	39
1.3.10. Пункт меню <i>Window</i> .....	40
1.3.11. Пункт меню <i>Help</i> .....	40
1.4. Основы работы с Derive 5 .....	42
1.4.1. Арифметические операторы .....	42
1.4.2. Алфавит системы .....	43
1.4.3. Ввод выражений .....	43
1.4.4. Вычисление функций .....	45
1.4.5. Создание функции пользователя .....	50

<b>Глава 2. Визуализация вычислений.....</b>	<b>53</b>
2.1. Двумерная графика .....	53
2.2. Трехмерная графика.....	63
<b>Глава 3. Вычисление математических функций .....</b>	<b>65</b>
3.1. Элементарные функции.....	65
3.1.1. Тригонометрические функции .....	65
3.1.2. Обратные тригонометрические функции.....	66
3.1.3. Гиперболические функции .....	67
3.1.4. Обратные гиперболические функции.....	69
3.1.5. Логарифмическая функция.....	70
3.1.6. Экспоненциальная функция .....	70
3.1.7. Функция извлечения корня квадратного из числа $x$ ..	71
3.2. Специальные функции.....	71
3.2.1. Математические функции.....	72
3.3. Вычисление функций при ограничивающих условиях .....	92
3.4. Системы счисления .....	95
3.4.1. Представление чисел в системах счисления.....	96
3.4.2. Перевод чисел из одной СС в другую .....	98
3.4.3. Арифметические операции в различных системах счисления.....	100
3.4.4. Решение задач в различных системах счисления....	101
3.5. Финансовые функции .....	104
<b>Глава 4. Специальные вычисления и преобразования математических функций.....</b>	<b>109</b>
4.1. Табулирование функции.....	110
4.2. Разложение функции в степенной ряд .....	116
4.2.1. Технология разложения функции в ряд Тейлора ....	116
4.2.2. Погрешности степенных рядов Тейлора.....	121
4.2.3. Компьютерные технологии оценки погрешностей рядов .....	122
4.3. Вычисление пределов.....	125
4.4. Вычисление суммы ряда.....	127
4.5. Вычисление произведения ряда.....	130
4.6. Вычисление производных .....	130

<b>Глава 5. Алгебра матриц и векторов .....</b>	<b>133</b>
5.1. Представление векторов и матриц на экране ПК.....	133
5.1.1. Набор вектора (матрицы) с клавиатуры .....	133
5.1.2. Использование пункта главного меню <i>Author</i> .....	134
5.2. Операции с векторами .....	136
5.3. Виды и характеристики матриц .....	138
5.3.1. Характеристики матриц .....	141
 <b>Глава 6. Решение алгебраических и трансцендентных уравнений .....</b>	 <b>145</b>
6.1. Решение уравнений с одним неизвестным .....	145
6.2. Решение уравнений в аналитическом виде.....	149
6.3. Численные методы решения алгебраических и трансцендентных уравнений.....	161
6.3.1. Метод дихотомии (половинного деления).....	166
6.3.2. Метод хорд .....	166
6.3.3. Метод касательных.....	168
6.3.4. Комбинированный метод (метод хорд и касательных) .....	170
6.3.5. Метод итераций .....	172
6.4. Компьютерные технологии решения уравнений .....	174
6.4.1. Определение области изоляции корня .....	175
6.4.2. Определение производных .....	176
6.4.3. Обеспечение сходимости итераций.....	177
6.4.4. Определение числа итераций .....	179
 <b>Глава 7. Решение систем алгебраических уравнений.....</b>	 <b>189</b>
7.1. Решение систем линейных алгебраических уравнений ....	190
7.1.1. Выбор начальных приближений .....	193
7.1.2. Условия сходимости итерационного процесса.....	194
7.1.3. Признак окончания вычислений .....	196
7.1.4. Алгоритмы метода итерации.....	196
7.2. Решение систем линейных уравнений с помощью системы Derive 5.....	198
7.2.1. Традиционные методы решения системы уравнений....	198

7.2.2. Матричный метод решения систем линейных алгебраических уравнений .....	208
7.3. Итерационные методы решения систем уравнений.....	213
7.4. Решение систем нелинейных уравнений .....	215
7.4.1. Метод Ньютона-Рафсона .....	215
7.4.2. Метод итераций .....	217
<b>Глава 8. Решение дифференциальных уравнений.....</b>	<b>229</b>
8.1. Постановка задачи.....	229
8.2. Приближенные аналитические методы решения дифференциальных уравнений .....	231
8.2.1. Метод последовательного дифференцирования .....	231
8.2.2. Метод неопределенных коэффициентов.....	232
8.2.3. Метод последовательных приближений .....	234
8.3. Аналитические методы решения дифференциальных уравнений в среде Derive 5 .....	234
8.3.1. Функции решения уравнений первого порядка.....	235
8.3.2. Функции решения уравнений второго порядка.....	241
8.3.3. Решение дифференциальных уравнений в аналитическом виде.....	245
8.4. Численные методы решения дифференциальных уравнений .....	248
8.4.1. Метод Эйлера.....	248
8.4.2. Усовершенствованные методы Эйлера .....	250
8.4.3. Метод Рунге-Кутты .....	252
8.5. Численные методы решения дифференциальных уравнений в среде Derive 5 .....	254
8.5.1. Функция <i>EULER</i> .....	254
8.5.2. Функция <i>TAY_ODE1</i> .....	256
8.5.3. Функция <i>PICARD</i> .....	257
8.5.4. Функция <i>RK</i> .....	258
<b>Глава 9. Вычисление интегралов .....</b>	<b>271</b>
9.1. Технология вычисления интегралов в системе Derive 5 .....	272
9.2. Алгоритмы численных методов вычисления интегралов ...	274
9.2.1. Формулы прямоугольников.....	274

9.2.2. Формула трапеций .....	275
9.2.3. Формула парабол (Симпсона) .....	275
9.3. Вычисление определенных интегралов по квадратурным формулам .....	276
9.3.1. Вычисление интеграла по методу парабол (Симпсона) .....	282
9.4. Вычисление кратных интегралов .....	287
9.5. Примеры вычисления интегралов .....	289

## **Глава 10. Интерполяция — метод моделирования .....297**

10.1. Этапы компьютерной технологии интерполяции .....	297
10.1.1. Выбор вида функции интерполяции .....	299
10.2. Интерполяция точная в узлах .....	303
10.2.1. Универсальный метод интерполяции точной в узлах .....	304
10.2.2. Функция <i>POLY_INTERPOLATE</i> .....	307
10.2.3. Функция <i>LAGRANGE</i> .....	308
10.2.4. Сплайн-интерполяция .....	310
10.3. Интерполяция приближенная в узлах (аппроксимация) ...	311
10.3.1. Функция <i>FIT</i> .....	311
10.3.2. Функция <i>ALL_SEVEN</i> .....	314
10.3.3. Паде-аппроксимация .....	316
10.4. Решение задач интерполяции путем прямых вычислений .....	317
10.5. Проверка адекватности модели .....	320

## **Глава 11. Задачи повышенной сложности .....327**

Задача 11.1 .....	327
Задача 11.2 .....	328
Задача 11.3 .....	329
Задача 11.4 .....	330
Задача 11.5 .....	331
Задача 11.6 .....	331
Задача 11.7 .....	332
Задача 11.8 .....	332
Задача 11.9 .....	333

---

Задача 11.10 .....	334
Задача 11.11 .....	334
Задача 11.12 .....	335
Задача 11.13 .....	335
Задача 11.14 .....	336
Задача 11.15 .....	336
Задача 11.16 .....	337
Задача 11.17 .....	337
Задача 11.18 .....	338
Задача 11.19 .....	338
Задача 11.20 .....	339





# Введение

Создание универсальных программных средств символьной математики стало, в последнее время, основой нового научного направления в информатике, которое получило название — *компьютерная алгебра*. В настоящее время наиболее известными и широко используемыми на практике являются такие математические системы, как Mathematica, Maple, Derive, Mathcad и др. Они позволяют решать задачи аналитическими и численными методами. Объем решаемых этими системами задач, а также время и точность их решения таковы, что поражают воображение самого взыскательного пользователя. Эти математические системы, в общей совокупности, представляют собой непревзойденные средства решения самых сложных математических задач. Интеллектуальная библиотека подобных систем должна стать достоянием ученого, студента, инженера — каждого, кто использует математику в своей деятельности. Надеемся, что и эта книга найдет свое место в такой библиотеке.

Особенностями современного образования стали массовость, высокая информативность и пониженный интерес студентов к знаниям. В условиях высокой массовости индивидуализация обучения практически невозможна, что явилось одной из основных причин посредственных знаний студентов. Необходимость высокой информативности привела к появлению в учебных планах вузов большого количества небольших по объему предметов. Студент получает знания типа "кое-что о многом". Основная причина пониженного интереса студента к знаниям — невостребованность знаний обществом. Считается, что в условиях рыночной экономики и бизнеса глубокие знания не нужны, достаточно четырех арифметических действий и умения работать на персональном

компьютере в среде так называемых офисных программ. По этой причине многие способные студенты учатся лишь на "удовлетворительно". Их знания можно охарактеризовать как "посредственные знания о многом".

Для повышения эффективности образования необходимо внедрять новые проблемные методы обучения. Среди них наиболее эффективными и реальными являются компьютерные методы.

Основой компьютеризации образования является применение обучающих программных средств и универсальных математических систем, относящихся к группе прикладных программ символьной математики. Интеллектуальная библиотека персонального компьютера — незаменимое средство активизации обучения по многим учебным дисциплинам вуза.

В связи с вышеизложенным, особое значение в учебном процессе приобретает система *Derive 5*. Она обладает следующими особенностями:

- ❑ относительно проста в изучении;
- ❑ поддерживает интерактивный режим общения (запрос-ответ) пользователя с персональным компьютером (ПК);
- ❑ не предъявляет высоких требований к типу и техническим характеристикам компьютера (реализуется практически на любом современном ПК);
- ❑ обладает элементами интеллектуальной системы, при решении задач символьной математики;
- ❑ обеспечивает высокую производительность;
- ❑ устойчива к ошибкам вычислительного процесса;
- ❑ обеспечивает высокую достоверность решения задач;
- ❑ самая дешевая из всех универсальных математических систем данного типа.

Опыт показывает, что *Derive 5*, по сравнению с другими системами, более легко внедряется в учебный процесс при изучении дисциплин широкого профиля — от естественно-научного цикла до специальных дисциплин.

Перечисленные достоинства Derive 5 не дают основания утверждать, что это самая лучшая математическая система. Она, как и всякая другая система, имеет и недостатки.

К недостаткам системы можно отнести следующие:

- ❑ довольно слабые графические возможности;
- ❑ не поддерживается анимация;
- ❑ поддерживается небольшое количество операций (несколько сот), относительно других систем, например, таких как Maple (около 3000 операций), Mathematica (около 1000).

Достоинства и недостатки системы во многом субъективные. Их почувствует пользователь при решении практических задач или примеров, которых в книге достаточно много.

## Особенности книги

Данная книга, по системе Derive 5, является первой на русском языке и существенно отличается от ее предыдущих версий. Главные отличия состоят в используемой технологии решения задач и форме организации диалога с пользователем, а также в числе используемых функций для решения математических задач.

В книге излагаются не только функции системы и примеры их реализации, как это делается в большинстве книг, посвященных описанию универсальных программных средств символьной математики, но также методы, алгоритмы и компьютерные технологии решения математических задач. Это избавляет пользователя от возможных ошибок при решении задач с помощью функций данной системы.

В книге рассматривается большое число задач, имеющих категорию — "сложные". Решение таких задач требует одновременного использования многих функций системы. Более того, их решение дает возможность, за относительно короткое время, глубоко изучить систему Derive 5.

В книге описаны элементы программирования на языке рассматриваемой системы. Кроме этого, приведены примеры программ

вычисления функций, решения уравнений и вычисления интегралов. Наличие многовариантных задач с ответами будет полезно преподавателям и студентам при выполнении домашних заданий, на лабораторном практикуме и при приеме зачетов.

## Для кого эта книга

Наиболее активным пользователем книги, очевидно, будет студент. Это учтено автором при описании методов и алгоритмов решения математических задач; приемов использования некоторых компьютерных технологий; содержания примеров и многовариантных задач для самостоятельного решения. Но это вовсе не значит, что книга по содержанию ограничена учебными программами вузовских предметов. В ней также описаны технологии решения задач с помощью функций и команд без каких-либо ограничений, в частности, связанных с вузовским обучением. Книга будет полезна ученому, инженеру, экономисту, любому специалисту, которому приходится решать задачи, связанные со сложными математическими вычислениями. А для лиц, которые только приступают к изучению и использованию программных средств символьной математики, рекомендуется начать именно с Derive 5.

# Глава 1



## Основы работы с системой Derive 5

В настоящее время нет стандарта на компьютерные технологии решения задач. Каждая система компьютерной алгебры имеет свой язык общения с пользователем.

При решении математических задач таким языком является алфавит, арифметические операторы, правила ввода математических выражений, набор функций и команд выполнения математических операций.

Данная глава посвящена описанию основ работы с системой Derive 5 при решении математических задач. В ней излагаются: алфавит системы, арифметические операторы, правила ввода математических выражений, встроенные функции и функции пользователя. Основное внимание уделяется изложению функций и команд решения математических задач, а также организации диалога пользователя с системой Derive 5.

Основы работы иллюстрируются решением примеров.

### 1.1. Derive 5 как универсальная система символьной математики

Компьютерная алгебра — это новое направление в информатике. Совсем недавно компьютер мог только вычислять. Он делал солидные расчеты по составленным заранее программам. Его возможности, в отношении символьных вычислений, были практически нулевыми. При этом, невозможно было выводить формулы,

вычислять неопределенный интеграл, брать производные, разлагать функцию в ряд, преобразовывать математические выражения и многое другое. Все это существенно ограничивало применение компьютера в научных исследованиях и в обучении.

Появление универсальных систем символьной математики существенно повысило роль компьютера не только в науке и образовании, но и во многих областях практической деятельности людей — инженерном деле, экономике, моделировании различных систем и процессов, планировании и обработке экспериментов и др. Компьютер стал интеллектуальным техническим средством.

Среди универсальных программных средств символьной математики наиболее известны — Mathematica, Maple, Derive, Mathcad, Matlab. Одним из признаков их интеллектуальности является количество функций символьной математики, которые поддерживаются этими системами. Среди них первые три наиболее интеллектуальны. Лидером является система Maple, имеющая около 3000 функций, затем Mathematica, содержащая более тысячи функций, и наконец Derive, в которой реализовано несколько сотен функций.

Системы Mathcad и Matlab имеют меньшее число функций, а их реализация вызывает у пользователя значительные трудности (особенно Matlab). Между тем они имеют такие возможности, которые не реализованы в системах Mathematica, Maple, Derive. Например, библиотеки Matlab содержат программы исследования динамики систем управления, планирования эксперимента, нечетких множеств и многое другое. Система Mathcad позволяет с высоким профессионализмом готовить документы (научные статьи, рефераты, отчеты и др.) с большим числом формул, математических расчетов и графиков. Практическое использование этих систем столь велико, что они стали наиболее популярными, а книги, посвященные их описанию, наиболее покупаемыми. Однако следует иметь в виду, что эти системы не могут заменить указанные выше, и наоборот.

Меньший спрос на первые три наиболее интеллектуальные системы объясняется тем, что научными исследованиями занимается

значительно меньшее количество людей, чем практическими расчетами.

Существует много хороших универсальных программных средств, но даже все они вместе взятые не могут заменить универсальные программные средства символьной математики. Вот один из наиболее ярких этому примеров. Программа Excel, называемая табличным процессором, позволяет производить различные вычисления над большими массивами чисел. Попробуйте решить с помощью Excel хотя бы одну из следующих задач:

1. Сложить два числа:  $\frac{2}{7} + \frac{3}{11} = ?$  Ответ:  $\frac{43}{77}$ .
2. Найти корни уравнений: а)  $x^5 - a = 0$ , б)  $x^2 + \ln(x) - x = -1$ .  
 Ответы:  $x_1 = a^{0,2}$ ,  $x_2 = a^{0,2}(-0,81 + 0,59i)$ ,  $x_3 = a^{0,2}(-0,81 - 0,59i)$ ,  
 $x_4 = a^{0,2}(0,31 - 0,95i)$ ,  $x_5 = a^{0,2}(0,31 + 0,95i)$ .
3. Вычислить интеграл:  $y = \int_0^{\infty} e^{-ax} dx$ . Ответ:  $\frac{1}{a}$ .

Функция представлена в виде таблицы:

x	1	2	3	4	5	6	7	8	9	10
y	7,7	21,25	42,2	70,1	104,6	145,6	192,8	246	305,4	370,6

4. Представить функцию в виде формулы.

Ответ:  $y = 2,5 + 5,2x^{1,85}$ .

5. Возвести в степень:  $a^0 = ?$ ,  $0^0 = ?$ .

Между тем большинство задач, решаемых Excel, можно также решить с помощью систем символьной математики.

Несмотря на то, что в системе Derive 5 в три, четыре раза меньше функций, чем в системах Mathematica и Maple, она обладает широкими возможностями символьных вычислений. Кроме этого, отличается простотой и исключительной ясностью диалога с пользо-

вателем, наличием превосходной системы помощи, большим числом примеров. К сожалению, все это выполнено только на английском языке.

Derive 5 является самой производительной и безаварийной из всех известных систем символьной математики, с высокой корректностью и достоверностью решений.

Диалог с пользователем Derive 5 реализует по следующей схеме:

- Ввод выражения.
- Команда действий.
- Ответ.

Например, пусть необходимо найти производную функции  $f(x)$ . Пользователь в этом случае вводит выражение функции, затем обращается к команде дифференцирования путем щелчка мыши по соответствующей кнопке и, после нажатия клавиши <Enter>, получает ответ. Пользователь самостоятельно не вводит команд в память компьютера и не пользуется клавиатурой (кроме ввода исходного математического выражения, функции или уравнения).

Такая схема решения математических задач обеспечивает пользователю максимум удобств и делает процесс решения задачи легким и похожим на компьютерную игру. Иногда, в сложных случаях, приходится набирать на клавиатуре и вводить в память компьютера команды. Но даже в этих случаях пользователь испытывает максимум удобств, благодаря автоматизации этих действий (например, подстановка данных в математические выражения, установка опций, ввод начальных значений и начальных условий и др.).

## 1.2. Главное окно системы

Главное окно системы Derive 5 показано на рис. 1.1. Оно состоит из нескольких уникальных строк, каждая из которых имеет свое особое назначение. Рассмотрим назначение каждой из них в отдельности.



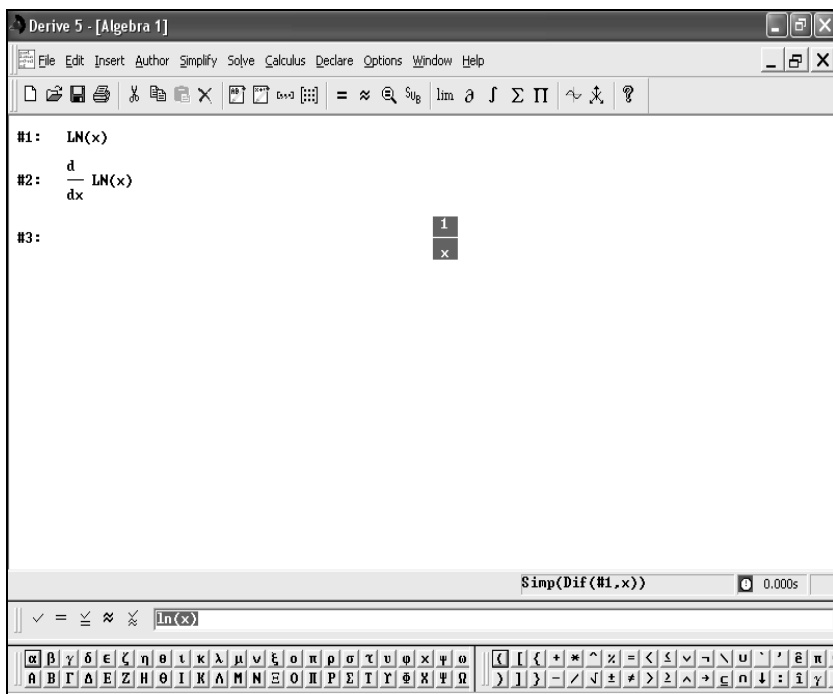


Рис. 1.1. Главное окно системы

Первая строка — *строка заголовка и управления окном*. Она расположена в верхней части главного окна и в ней находится название системы — Derive 5, а также кнопки управления окном (свертывания, развертывания и закрытия).

### Примечание

На рисунке кнопки управления окном не показаны.

Ниже строки заголовка расположена *строка главного меню*. Эта строка содержит следующие пункты:

- ☐ **File** — работа с файлами и принтером;
- ☐ **Edit** — редактирование выражений;

- ❑ **Insert** — открытие графических окон, аннотация к графикам;
- ❑ **Author** — ввод математических выражений и текстов для их дальнейшего использования;
- ❑ **Simplify** — преобразование и вычисление математических выражений;
- ❑ **Solve** — решение уравнений и систем уравнений;
- ❑ **Calculus** — вычисление производных, интегралов, пределов, сумм, произведений, разложение в ряд Тейлора;
- ❑ **Declare** — задание функций и переменных;
- ❑ **Options** — задание необходимых установок системы Derive;
- ❑ **Window** — открытие окон и работа с ними;
- ❑ **Help** — включение справочной системы.

Активизация пунктов главного меню осуществляется по щелчку левой кнопки мыши с предварительной установкой ее указателя в область выбранной позиции меню. Возможен вариант использования "горячих" клавиш. В этом случае используется сочетание клавиши <Alt> и клавиши, помеченной буквой, которая соответствует подчеркнутой букве в названии команды. Эти способы применяются и для управления разными командами, для которых "горячие" клавиши указываются после имени команды.

Главное меню изменяется в зависимости от состояния системы. Так, например, если в системе закрыты все окна, то главное меню будет иметь только пункты **File**, **Window** и **Help**. Графическое окно имеет свое меню. В ряде случаев некоторые пункты главного меню недоступны пользователю (надписи в них сделаны затененным шрифтом) или вовсе отсутствуют.

Следующая строка — *панель инструментов*. Панель инструментов расположена в третьей сверху экранной строке. Она содержит кнопки быстрого управления системой, дублирующие команды главного меню. Каждая из кнопок вводит из множества команд главного меню лишь одну. Набор кнопок зависит от вида окна, открытого в данный момент.

Панель инструментов дублирует многие важные и часто используемые команды. При наличии этой панели (по желанию ее можно убрать с экрана) во многих случаях можно не обращаться к пунктам и командам главного меню, используя одноименные кнопки панели инструментов). Наиболее часто приходится пользоваться кнопками быстрого управления при работе с математическими выражениями.

Ниже приводятся названия кнопок, классифицированных по группам:

□ Команды работы с файлами:

- **New** — открытие нового окна;
- **Open** — вывод окна загрузки файла;
- **Save** — сохранение данных в файле под текущим именем;
- **Print** — печать содержимого окна.

□ Команды редактирования:

- **Gut** — удаление выделенного выражения (множества выделенных выражений);
- **Copy** — создание копии;
- **Paste** — восстановление последнего удаленного выражения;
- **Delete Object** — удаление выделенного выражения без его восстановления.

□ Команды ввода:

- **Insert Text** — создание аннотаций;
- **Author Expression** — ввод математических выражений;
- **Author Vector** — задание вектора необходимой размерности;
- **Author Matrix** — задание матрицы необходимой размерности.

□ Команды вычислений:

- **Simplify** — символьные и точные вычисления в цифровой форме, упрощение математических выражений;

- **Approximate** — вычисления в цифровом виде с представлением чисел в естественной форме;
- **Solve Expression** — решение уравнений и систем уравнений;
- **Variable Substitution** — подстановка значений переменных.

□ Команды специальных вычислений:

- **Find limit** — вычисление пределов функций;
- **Find Derivative** — вычисление производных;
- **Find Integral** — вычисление интегралов;
- **Find Sum** — вычисление сумм рядов;
- **Find Product** — вычисление произведений рядов.

□ Команды графических окон:

- **2D-plot window** — вывод окна двумерной графики;
- **3D-plot window** — вывод окна трехмерной графики;
- **Help About Derive** — информация о системе Derive.

За рассмотренными тремя экранными строками располагается *окно выражений*. Оно занимает большую часть экрана и размещается сразу за панелью инструментов. В этом окне находятся функции, математические выражения и другая информация. Строки в "окне выражений" нумеруются, благодаря чему имеется возможность выполнять математические действия над выражениями посредством номеров строк, в которых они находятся. Выделение строки осуществляется щелчком кнопки мыши в ее области. Над выделенными выражениями осуществляются действия путем команд главного меню или панели инструментов.

Следующим атрибутом интерфейса системы является *диалоговое окно ввода выражений*.

Диалоговое окно активизируется (мигающий курсор в окне) командой **Author | Expression**, или нажатием клавиши <F2>, или щелчком левой кнопки мыши в область диалогового окна. Это окно служит для ввода математических выражений и текстов.

### Примечание

Редактирование выражения, уже введенного и отображаемого в основном окне, осуществляется только путем его вызова в диалоговое окно, с последующим вводом как нового. Редактирование осуществляется аналогично тому, как это делается в среде Windows.

К диалоговому окну добавлена панель с математическими символами и символами латинского и греческого алфавитов. Для ввода любого из символов достаточно указать на него курсором и щелкнуть левой кнопкой мыши. При этом курсор диалогового окна следует установить в месте ввода символа.

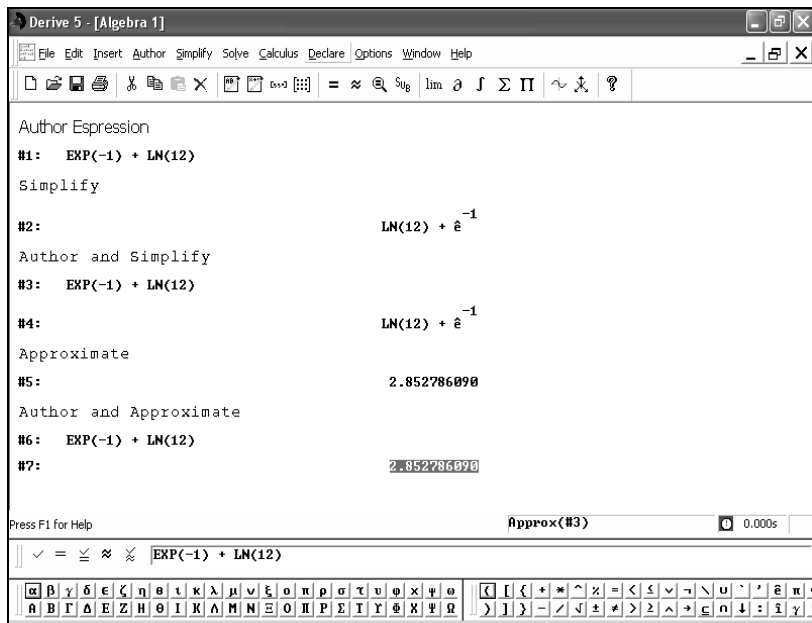
Слева от диалогового окна расположены пять кнопок. Они позволяют получать решения в различном виде без предварительного ввода вычисляемого выражения. Кнопки имеют следующие имена и назначения:

- ☐ **Author Expression** — ввод выражения (кнопка, дублирующая клавишу <Enter>);
- ☐ **Simplify** — упрощение вводимого выражения или его вычисление;
- ☐ **Author and Approximate** — ввод и вычисление выражения (на экране появляется выражение и его вычисленное значение).

Например, если в диалоговом окне задано выражение  $\exp(-1) + \ln(12)$ , то после нажатия соответствующих кнопок на экране будут следующие выражения:

Author Expression	
#1: $\exp(-1) + \ln(12)$	
Simplify	
#2 :	$\ln(12) + e$
Author and Simplify	
#3: $\exp(-1) + \ln(12)$	
#4	$\ln(12) + e$
Approximate	
#5	2.852786
Author Approximate	
#6: $\exp(-1) + \ln(12)$	
#7:	2.852786

Вид экрана показан на рис. 1.2.



**Рис. 1.2.** Окно решения задачи без предварительного ввода выражения

Последней строкой является *строка состояния системы*. Она расположена в нижней части основного окна (на рис. 1.1 отсутствует) и предназначена для оперативного контроля за работой системы. В ней выводятся текстовые комментарии о действиях, выполняемых над выражениями, пояснения о выполняемых командах, сообщения о времени выполнения расчетов и правильности их завершения, а также о параметрах графиков для графических окон.

## 1.3. Главное меню системы

### 1.3.1. Пункт меню *File*

Пункт **File** имеет ряд команд, определяющих его подменю.

- ☐ **New** (<Ctrl>+<N>) — удаление содержимого окна выражений и создание нового окна.

- ❑ **Open** (<Ctrl>+<O>) — вызов окна файлов и их загрузка.
- ❑ **Close** — закрытие окна.
- ❑ **Safe** (<Ctrl>+<S>) — вызов окна записи файла с целью его сохранения.
- ❑ **Safe As** — создание файла с присвоением ему уникального имени.
- ❑ **Load** — обращение к подменю загрузки файлов.
- ❑ **Write** — запись файла в формате различных языков программирования.
- ❑ **Pade Setup** — установка параметров печати.
- ❑ **Print Preview** — предварительный просмотр содержимого печати.
- ❑ **Print** (<Ctrl>+<P>) — вызов диалогового окна печати.
- ❑ **Exit** — выход из среды Derive.

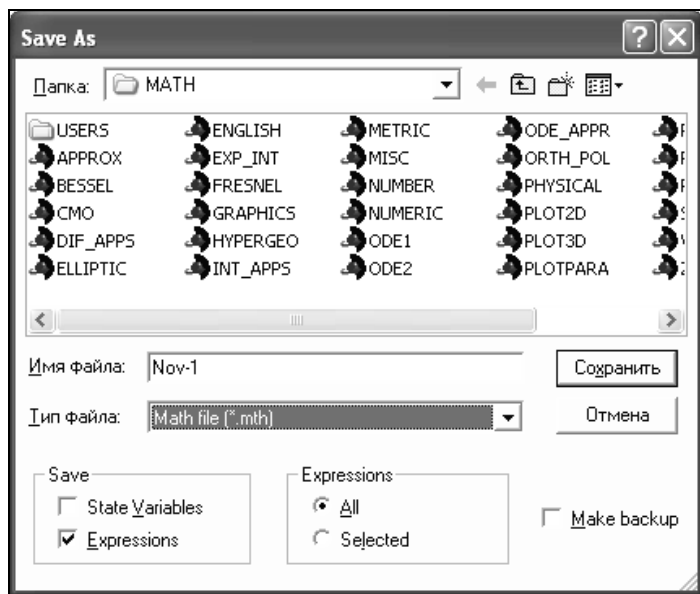
### Примечание

Команды подменю **Load** и **Write** имеют свои подменю с целым рядом команд, отмеченных треугольниками.

Рассмотрим назначение команд пункта **File** главного меню.

- ❑ **New** (<Ctrl>+<N>) — создание нового окна. Выполнение этой команды приводит к полному уничтожению всех открытых окон и создает новое окно с именем — ???МТН. При этом исчезает также выражение, которое находится в диалоговом окне.
- ❑ **Open** (<Ctrl>+<O>) — команда вызывает новое окно, позволяющее выбирать на обработку любой файл с жесткого или гибкого диска.
- ❑ **Close** — команда закрытия экранного окна. При исполнении команды закрывается только одно окно, остальные остаются на экране. При закрытии система предлагает сохранить содержимое этого окна, если ранее оно не сохранялось на диске.
- ❑ **Save** (<Ctrl>+<S>), **Safe As** — команды сохранения. Команда **Save** служит для сохранения файла с его текущим именем.

Команда **Safe As** позволяет сохранить файл с его уникальным именем. При обращении к ней появляется диалоговое окно сохранения файла (см. рис. 1.3).



**Рис. 1.3.** Окно сохранения файла

Чтобы выполнить операцию, надо выполнить следующие действия:

- в строке ввода **Имя файла** записать имя файла с расширением;
- нажать кнопку **Сохранить**.

### Примечание

Расширение можно не вводить, программа выберет его самостоятельно.

Внизу окна **Save As** расположены две панели — **Save** и **Expressions**, которые определяют опции записываемого файла. На панели **Save** определяются два флажка с именами **State Variables** и **Expressions**, которые позволяют задавать статус



переменных и запись выражений путем установки знака "V" (щелчок мыши в области прямоугольника). На панели **Expressions** определяется флажок, который может принимать одно из двух альтернативных значений с именами — **All** и **Selected**. Если программа позволяет записать в файл все выражения, то значение флажка устанавливается **All**. Если только выделенные, то — **Selected**.

### Примечание

Для этого необходимо щелкнуть левой кнопкой мыши по нужному кружку. В кружке образуется точка.

Команда **Write** имеет подменю, включающее следующие команды:

- ☐ **Basic File** — запись выражений в формате языка Бейсик;
- ☐ **C File** — запись выражений в формате языка Си;
- ☐ **Fortran File** — запись выражений в формате языка Фортран;
- ☐ **Pascal File** — запись выражений в формате языка Паскаль;
- ☐ **Rich Text Format File** — сохранение файла выражений.

Эти команды вызывают диалоговое окно, подобное окну **Save As**, и дают возможность записать математические выражения, которые могут быть необходимыми при совместном функционировании Derive и универсальных языков программирования.

Подменю **Load** служит для вызова и выполнения следующих команд:

- ☐ **Math File** — вызов диалогового окна с выражениями;
- ☐ **Data File** — вызов диалогового окна загрузки файла данных;
- ☐ **Demo File** — вызов диалогового окна демонстрационных файлов;
- ☐ **Utility File** — вызов диалогового окна загрузки утилит.

Команда **Math** служит для загрузки математических файлов с расширением **mtx** или любым другим расширением.

### Примечание

На рис. 1.3 показано окно файлов с расширением **mtx**.

После выбора нужного файла, путем его выделения и загрузки, в окне математических выражений появляется его содержимое. Содержимое файла располагается в очередной строке и имеет номер. Им можно пользоваться, как и любым другим выражением, находящимся в окне выражений. Используя вкладку **Тип файлов** можно, путем переключения, вызвать файлы с другим расширением.

Команда **Data** служит для загрузки файлов данных. Это дает возможность пользоваться внешними данными в процессе решения математических задач.

Команда **Demo** служит для загрузки демонстрационных файлов. Она выводит окно демонстрационных файлов, которое, в свою очередь, позволяет вызвать любой из имеющихся в нем файлов.

Команда **Utility** служит для вызова утилит, входящих в библиотеки расширений системы. Загрузка осуществляется в память компьютера. На экран содержимое файла не вызывается, вызывается лишь имя утилиты, которое является информацией о загрузке данной утилиты.

Команда **Pade Setup** позволяет установить требуемые параметры печати страницы. Окно установки параметров печати показано на рис. 1.4.

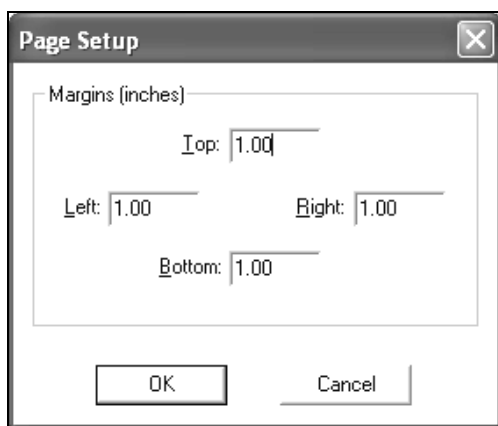


Рис. 1.4. Окно установки параметров печати страницы

Команда позволяет начать печатать непосредственно из окна предварительного просмотра.

Команда **Print** выводит на экран окно печати (рис. 1.5), в котором устанавливаются параметры печати: число копий и диапазон страниц.

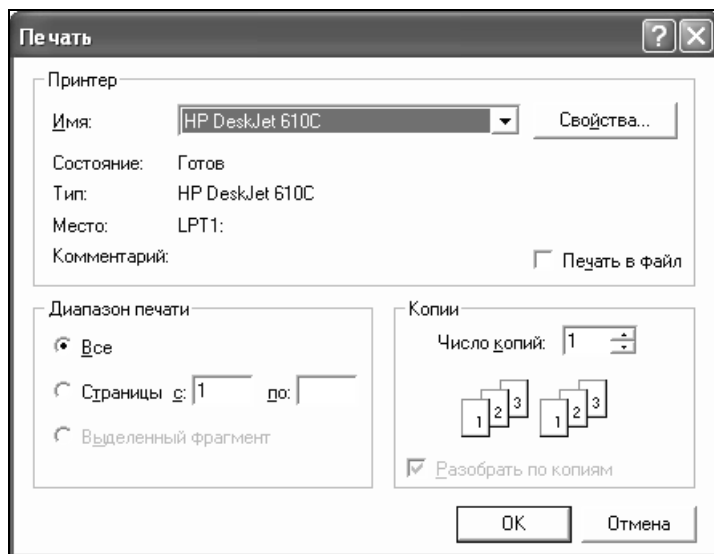


Рис. 1.5. Окно установки параметров печати

Команда **Exit** (рис. 1.6) служит для выхода из системы Derive 5. При обращении к этой команде появляется диалоговое окно, в котором пользователю предлагается: сохранить документ (**Да**), не сохранять (**Нет**), продолжить работу в системе (**Отмена**).

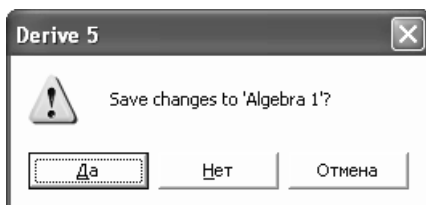


Рис. 1.6. Окно выхода из Derive 5

### 1.3.2. Пункт меню *Edit*

Редактирование выражений и текста документа осуществляется с помощью пункта главного меню — **Edit**. Редактор **Edit** содержит следующие команды:

- ☐ **Derive Object** (<Enter>) — выделение выражения специальным знаком;
- ☐ **Annotation** — вывод окна аннотации выделенного выражения;
- ☐ **Delete** — удаление выделенного выражения;
- ☐ **Undelete** (<Ctrl>+<Z>) — восстановление последнего удаленного выражения;
- ☐ **Select All** (<Ctrl>+<A>) — выделение всех выражений экрана;
- ☐ **Cut** (<Ctrl>+<X>) — удаление выделенных выражений;
- ☐ **Copy** (<Ctrl>+<C>) — копировать выделенные выражения в буфер обмена;
- ☐ **Paste** (<Ctrl>+<V>) — восстановление выражений из буфера обмена;
- ☐ **Mark and Copy** — выделение и копирование выражений.

Рассмотрим назначение команд редактора.

Команда **Derive Object** выделяет объект, помещая его в рамку. Вместо выделенного выражения можно поместить новое, при этом номер строки не меняется. Замена выражения осуществляется путем набора нового выражения и нажатия клавиши <Enter>.

Команда **Annotation** выводит окно, в котором можно набрать текст и поместить на экран в качестве аннотации к выделенному выражению.

Команда **Delete** удаляет с экрана все выделенные выражения. При этом выражение, находящееся в диалоговом окне ввода выражений, не удаляется. Эту команду дублирует кнопка **Delete Object** на панели инструментов.

Команда **Undelete** восстанавливает последнее удаленное выражение (выражения). Восстановление осуществляется столько раз,

сколько раз выполняется эта команда. Восстановление предыдущего удаленного выражения с помощью этой команды невозможно.

Команда **Sellect All** выделяет все выражения экрана.

Буфер обмена операционной системы Windows, именуемый Clip-board, является областью памяти, предназначенной для хранения информации. С помощью этого буфера осуществляется обмен информацией между различными приложениями, которые выполняются под управлением Windows. Общение Derive с буфером обмена осуществляется с помощью следующих команд: **Cut**, **Copy**, **Paste**, **Mark and Copy**.

Команда **Cut** удаляет выражение или его часть без сохранения в буфере обмена. Команду дублирует кнопка **Cut** панели инструментов.

Команда **Copy** копирует выделенные выражения в буфер обмена, оставляя их на экране.

Команда **Paste** осуществляет восстановление скопированных в буфер обмена выражений.

Команда **Mark and Copy** выделяет и копирует выражения в буфер обмена.

Последние скопированные выражения можно вызвать на экран с помощью команды **Paste** даже после нового сеанса работы системы Derive. После исполнения этой команды на экране появляется курсор мыши в виде креста. Если теперь нажать левую кнопку мыши и перемещать ее, то появится прямоугольник, который при движении мыши расширяется. Это позволяет выделить любой участок экрана с выражениями. Отпустив левую кнопку мыши, выделенный участок будет скопирован в буфер обмена, в чем можно убедиться, исполнив команду **Paste**.

### 1.3.3. Пункт меню *Insert*

Пункт главного меню **Insert** открывает новые объекты, определяемые следующими командами:

- **2D-plot Object** — открытие окна двумерной графики (дублирует команда 2D-plot window панели инструментов);

- ❑ **3D-plot Object** — открытие окна трехмерной графики (дублирует команда **3D-plot window** панели инструментов);
- ❑ **Text Object** — открытие строки текста, при этом пользователь имеет возможность вводить текстовую информацию, в том числе и на русском языке;
- ❑ **OLE Object** — добавление в документ нового объекта.

### 1.3.4. Пункт меню *Author*

Пункт главного меню **Author** имеет подменю со следующими командами:

- ❑ **Expression** (<Ctrl>+<A>) — ввод математических выражений;
- ❑ **Vector** — определение размеров вектора;
- ❑ **Matrix** — задание размеров матрицы.

Команда **Expression** ведет к появлению курсора в окне ввода выражений, который разрешает вводить и редактировать математические выражения. Для ввода специальных математических знаков можно использовать знаковую панель, которая всегда в распоряжении пользователя. Достаточно курсор установить в нужном месте выражения и щелкнуть мышью по требуемому знаку.

Команда **Vector** вызывает окно **Vector Setup**, в котором устанавливается размер вектора (рис. 1.7).

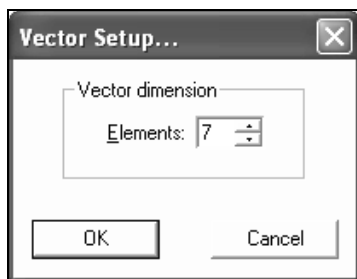
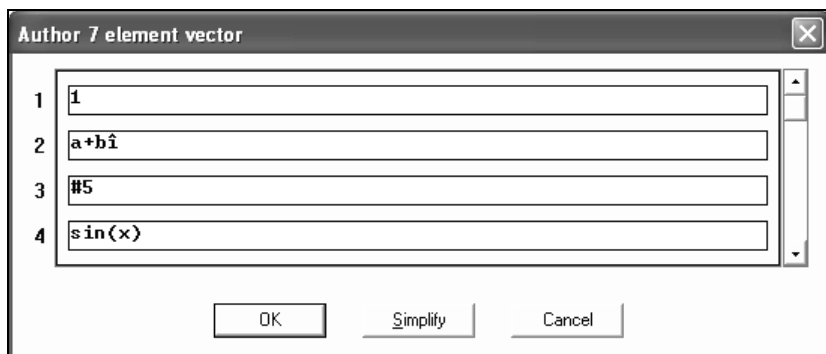


Рис. 1.7. Окно ввода размера вектора

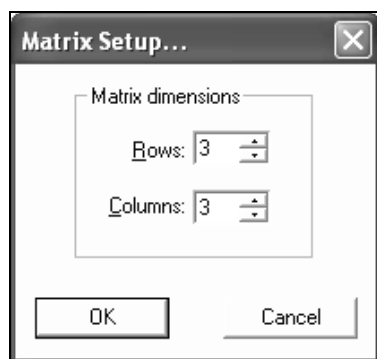
После ввода числа элементов вектора и щелчка мышью по кнопке **OK** появляется новое окно с именем **Author 7 element ?**. Вопросительный знак означает имя вектора с расширением MTH (рис. 1.8).



**Рис. 1.8.** Окно ввода элементов вектора

Элементами вектора могут быть — числа; символьные переменные; выражения; формулы; номера строк, в которых находятся выражения, и т. д. На рис. 1.8 показано окно ввода, в которое помещены: вещественное число, комплексное выражение, номер строки некоторого выражения и функция.

Команда **Matrix** вызывает окно **Matrix Setup** (рис. 1.9).



**Рис. 1.9.** Окно ввода размеров матрицы

В окне **Matrix Setup** устанавливается размер матрицы — число строк (**Rows**) и число столбцов (**Columns**). После их установки (непосредственно или с помощью счетчика строк и столбцов, с последующим щелчком мышью по кнопке **OK**) на экране появляется окно ввода элементов матрицы: **Author 3 x 3 matrix** (рис. 1.10).

	1	2	3
1	1	2	3
2	3	0	5
3	3	7	1

OK      Simplify      Cancel

**Рис. 1.10.** Окно ввода элементов матрицы

В нашем случае вводится матрица  $3 \times 3$ . Элементами матрицы, так же как и вектора, могут быть числа, символьные переменные, выражения и т. д.

### 1.3.5. Пункт меню *Simplify*

Пункт меню **Simplify** имеет следующие команды:

- ☐ **Basic** (<Ctrl>+<B>) — упрощение выражений общего вида;
- ☐ **Expand** (<Ctrl>+<E>) — расширение выражений;
- ☐ **Factor** (<Ctrl>+<F>) — факторизация выражений;
- ☐ **Approximate** (<Ctrl>+<G>) — вычисление выражений или их отдельных частей.

#### Примечание

Если в выражении численных значений нет, то команда **Approximate** упрощает выражение, причем ее дублирует команда **Approximate** ( $\approx$ ) на панели инструментов.



Наиболее важными здесь являются команды **Expand** и **Factor**.

Команда **Expand** называется командой расширения. Она позволяет раскрывать скобки в алгебраических выражениях, осуществлять разложение на элементарные дроби, выполнять сокращение дробей. Технология этих преобразований проста и состоит в выполнении нескольких действий.

- Ввод выражения.
- Выполнение команды **Simplify** | **Expand** (на экране формируется окно **Expand Expression**).
- Активизация необходимых переменных в окне **Expression Variables** и щелчок мыши по одной из кнопок окна **Amount**.
- Щелчок мыши по кнопке **Expand** или **OK** (в первом случае на экране появится ответ, во втором — команда, причем для ее реализации необходимо щелкнуть мышью по кнопке **Simplify**, расположенной на панели инструментов).

В окне **Amount** помечены четыре опции команды **Expand**: **Trivial** (тривиальное), **Square Free** (свободное от радикалов), **Rational** (рациональное), **Radical** (радикальное). Выбор типа выражения осуществляется щелчком кнопки мыши по соответствующей опции.

Ниже приводятся примеры преобразования выражений с помощью команды **Expand**.

*Выражение исходное*

*Выражение после команды*

$$(x + a)^5$$

$$x^5 + 5ax^4 + 10a^2x^3 + 10a^3x^2 + 5a^4x + a^5$$

$$\frac{x^3 - a^3}{x - a}$$

$$x^2 + ax + a^2$$

$$\frac{x^3 - a}{x - a}$$

$$\frac{a(a^2 - 1)}{x - a} + x^2 + ax + a^2$$

$$\frac{124}{16}$$

$$\frac{31}{4}$$

$$(x - 1)^2(x^3 - 2)$$

$$x^5 - 2x^4 + x^3 - 2x^2 + 4x - 2$$

Команда **Factor** дает возможность выполнения операций разложения многочлена на множители, выноса общего члена за скобки, приведения дроби к общему знаменателю. Так же, как и команда **Expand**, **Factor** имеет следующие опции: **Trivial**, **Square Free**, **Rational**, **Radical**, **Complex**.

При наличии в команде опции **Trivial** выражения приводятся к общему знаменателю, осуществляется вынос общего члена за скобки. При разложении, свободном от радикалов (**Square Free**), выполняются те же операции, что и при наличии опции **Trivial**. Отличие состоит лишь в различных формах представления результатов. При разложении с опцией **Rational** в конечном выражении отсутствуют радикалы. При наличии в команде **Factor** опции **Radical** осуществляется разложение с возможностью представления чисел с дробными степенями. При наличии опции **Complex** в разложении используются комплексные числа.

Технология преобразований с помощью команды **Factor** не отличается от техники использования команды **Expand**.

Ниже приведены примеры преобразования выражений с помощью команды **Factor**.

*Выражение исходное*

*Выражение факторизованное*

$$\frac{a}{b} + \frac{c}{a+b}$$

$$\frac{a^2 + ab + cb}{b(a+b)}$$

$$ax^2 + bx$$

$$x(ax+b)$$

$$9x^3 + 22,5x^2 + 0,5x - 7$$

$$\frac{(2x-1)(3x+2)(3x+7)}{2}$$

*или при опции **Radical***

$$\frac{9(x + \frac{2}{30})(x + \frac{7}{3})(2x-1)}{2}$$

$$a^2 + a - 1$$

$$(a + \frac{\sqrt{5}}{2} + \frac{1}{2})(a - \frac{\sqrt{5}}{2} + \frac{1}{2})$$

### Упражнения.

С помощью команд **Expand** и **Factor** преобразуйте следующие выражения:

$$\frac{2}{x+a} - \frac{3}{x+a} + \frac{1}{x^2 - a^2}, \quad x^5 - 2, \quad (x+1,2)(x-3,6)(x+7),$$

$$\frac{(\sin(x)^2 + \cos(x)^2)\cos(x)}{\sin(2x)}, \quad \frac{(a+b)^2 - c^2}{a+b+c}, \quad x^4 - 3x^3j + 3xj + 9x^2 - 10.$$

Проверьте правильность преобразования, используя те же команды **Expand** и **Factor**, а также их опции.

## 1.3.6. Пункт меню **Solve**

Пункт главного меню **Solve** имеет две команды:

- **Expression** (<Ctrl>+<Chift>+<E>) — решение уравнений с одним неизвестным;
- **Sistem** (<Ctrl>+<Chift>+<Y>) — решение систем уравнений.

Команда **Expression**, при ее исполнении, приводит к появлению окна **Solve Expression**, имеющего четыре вставки. Команду дублирует кнопка **Expression** панели инструментов.

### Примечание

Подробно команда рассмотрена в гл. 6.

Команда **Sistem**, при ее исполнении, приводит к появлению окна **Solve1 System Setup**, в котором устанавливается порядок системы уравнений. После нажатия кнопки **ОК** появляется новое окно — **Solve2 equation**, для ввода уравнений.

### Примечание

Технология решения уравнений рассмотрена в гл. 7.

### 1.3.7. Пункт меню *Calculus*

Пункт главного меню **Calculus** имеет следующие команды:

- ❑ **Limit** (<Ctrl>+<Shift>+<L>) — определение пределов функции;
- ❑ **Differentiate** (<Ctrl>+<Shift>+<D>) — вычисление производных;
- ❑ **Taylor Series** (<Ctrl>+<Shift>) — разложение функции в ряд Тейлора;
- ❑ **Integrate** (<Ctrl>+<Shift>+<I>) — определение значения интеграла;
- ❑ **Sum** (<Ctrl>+<Shift>+<S>) — вычисление суммы членов ряда;
- ❑ **Product** (<Ctrl>+<Shift>+<P>) — вычисление произведения членов ряда;
- ❑ **Vector** (<Ctrl>+<Shift>+<R>) — вычисление значений функции и представление ее в виде вектора;
- ❑ **Table** (<Ctrl>+<Shift>+<A>) — вычисление значений функции и представление ее в виде таблицы.

Команды **Limit**, **Differentiate**, **Integrate**, **Sum**, **Product** имеют дублирующие кнопки на панели инструментов (соответственно —  $\lim$ ,  $\partial$ ,  $\int$ ,  $\Sigma$ ,  $\Pi$ ).

#### Примечание

Подробное описание функций дано в гл. 4.

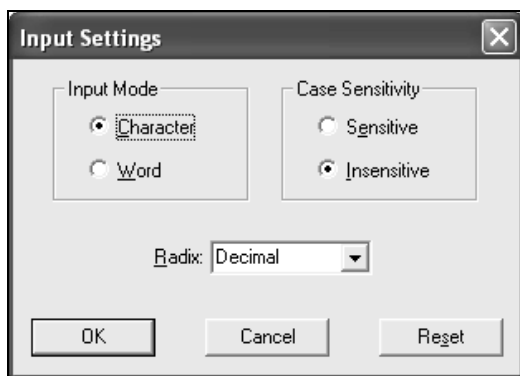
### 1.3.8. Пункт меню *Declare*

Пункт главного меню **Declare** (декларация переменных и функций) имеет следующие команды:

- ❑ **Variable Value** (<Ctrl>+<Alt>+<V>) — присвоение переменным численных значений;

- ❑ **Variable Domain** (<Ctrl>+<Alt>+<D>) — определение области значений переменной;
  - ❑ **Function Definition** (<Ctrl>+<Alt>+<F>) — установление имени функции;
  - ❑ **Input Settings** (<Ctrl>+<Alt>+<I>) — вывод окна опций управления форматом ввода данных;
  - ❑ **Output Settings** (<Ctrl>+<Alt>+<O>) — вывод окна опций управления форматом вывода данных;
  - ❑ **Simplification Settings** (<Ctrl>+<Alt>+<S>) — вывод окна упрощения
- Reset All Settings** (<Ctrl>+<Alt>+<R>) — отмена всех изменений.

Команда **Input Settings** создает окно с тем же именем и двумя панелями — **Input Mode** и **Case Sensitivity** (рис. 1.11).



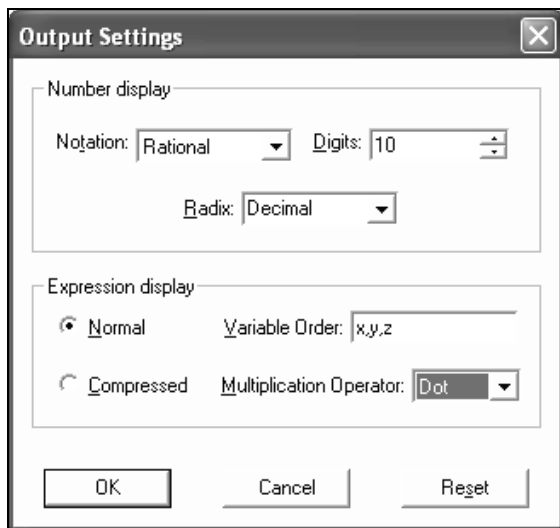
**Рис. 1.11.** Окно установки переменных

**Input Mode** служит для установления имени переменных. Запись  $x1$  можно воспринимать как одно имя переменной (**Word**) или как произведение  $x \times 1$  (**Character**). **Case Sensitivity** устанавливает чувствительность к регистрам переключения символов. По умолчанию устанавливается опция **Insensitive**, что означает нечувствительность к регистрам.

В окне имеется опция установления системы счисления (**Radix**): **Decimal** — десятичная, **Binary** — двоичная, **Octal** — восьмеричная,

**Hexadecimal** — шестнадцатеричная. По умолчанию установлена десятичная система счисления. Три нижних переключателя служат для ввода установленного режима (**OK**), отмены команды **Input Settings (Cancel)**, восстановления установленных по умолчанию опций (**Reset**).

Команда **Output Settings** создает окно того же имени с двумя панелями: **Number display** и **Expression display** (рис. 1.12).

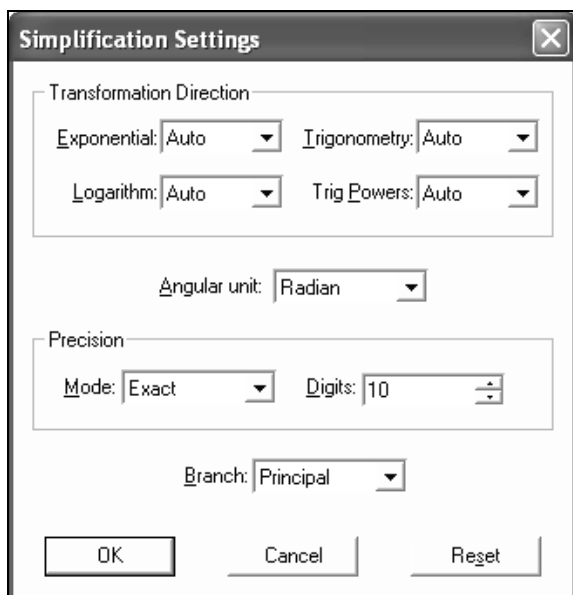


**Рис. 1.12.** Окно формата вывода данных

Панель **Number display** служит для установки и отображения текущего формата чисел: (**Decimal** — десятичный, **Mixed** — смешанный, **Rational** — рациональный), количества цифр числа (**Digits**), основания системы счисления (**Radix**).

Панель **Expression display** используется для определения вывода выражений — в нормальной (**Normal**) и сжатой (**Compressed**) формах, а также для установления порядка переменных (**Variable Order**) и знака операций умножения (**Multiplication Operator**): **Asterisk** (\*), **Dot** (•), **Implicit** (пробел).

Команда упрощения **Simplification Settings** выводит окно того же имени (рис. 1.13) с четырьмя областями управления процессом выполнением этой команды: **Transformation Direction** (направления преобразования), **Angular unit** (представление узлов), **Precision** (точность), **Branch** (выбор ветви преобразования).



**Рис. 1.13.** Окно команды упрощения

### 1.3.9. Пункт меню *Options*

Пункт главного меню **Options** позволяет установить опции параметров текста, способов размещения информации на экране, параметров печати, номеров строк математических выражений, цвета объектов и др.

Этот пункт меню имеет следующие команды: **Display**, **Printing**, **Startup**, **Renumberr Expression**, **Hide Labels**, **Hide Plots**, **Hide Text**, **Hide OLE Objects**. Все перечисленные команды дают возможность создать документ высокого качества.

**Примечание**

В нашу задачу не входит изложение вопросов создания рабочего места математика на базе Derive 5. Читателю будет не трудно самостоятельно, решая конкретную задачу, применить команды пункта **Options** для создания качественного текста.

### 1.3.10. Пункт меню *Window*

При большом числе выражений весь документ не виден. Кроме того, трудно найти необходимую строку, если их в документе большое количество. Поэтому в Derive предусмотрена возможность организации работы с использованием многооконной системы отображения. В этом случае окна можно изменять в размерах и перемещать по экрану. Эти функции выполняет пункт главного меню **Window**, который содержит следующие команды:

- ☐ **Cascade** (<Ctrl>+<Shift>+<C>) — каскадное расположение окон;
- ☐ **Tile Horisontally** (<Ctrl>+<Shift>+<H>) — горизонтальное расположение окон;
- ☐ **Tile Vertically** (<Ctrl>+<Shift>+<V>) — вертикальное расположение окон;
- ☐ **Display Tabs** (<Ctrl>+<Shift>+<T>) — показать закладки.

Кроме перечисленных команд, пункт меню **Window** создает окна двумерной и трехмерной графики. Это можно сделать с помощью команд:

- ☐ **New 2D-PlotWindow** — создание окна двумерной графики;
- ☐ **New 3D-PlotWindow** — создание окна трехмерной графики;
- ☐ **View Toolbars** — установка или удаление панели инструментов.

### 1.3.11. Пункт меню *Help*

Пункт главного меню **Help** обеспечивает пользователя справочной системой, содержащей сведения о всех командах и функциях



и имеет большое количество примеров. Пункт меню **Help** поддерживает следующие команды:

- ☐ **Contents** — вызов контекстной справки, позволяющей найти по наименованию темы нужную информацию (например, назначение команды, структура функции и т. д.);
- ☐ **Index** — вызов индексного каталога, позволяющего в алфавитном порядке выбрать нужное слово и получить справку о нем, выделив это слово и нажав клавишу <Enter>;
- ☐ **Frequently Asked Questions** — часто спрашиваемые вопросы и ответы;
- ☐ **Additional Resources** — дополнительные возможности;
- ☐ **Link to Derive Web Site** — связь с Интернетом;
- ☐ **About Derive** — вывод окна с краткой справкой о системе Derive 5 (рис. 1.14).

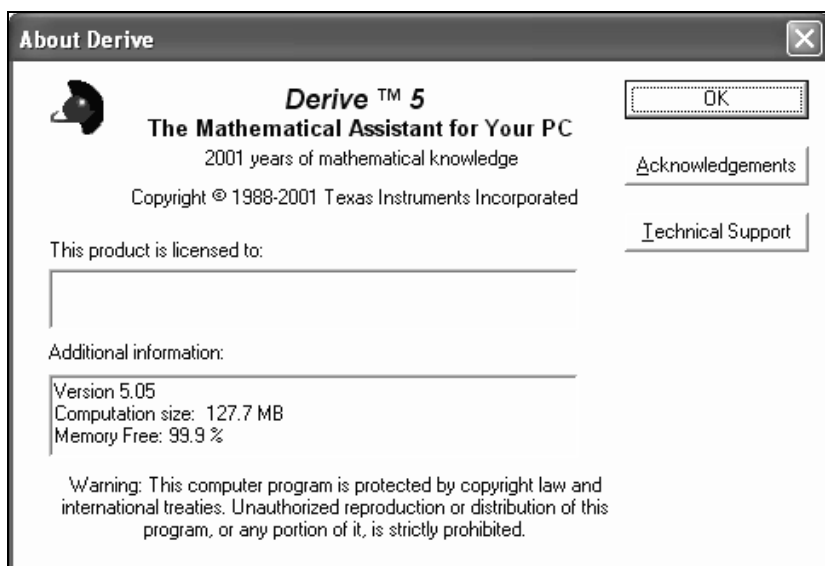


Рис. 1.14. Окно справки о Derive 5

Из рис. 1.14 видно, что система Derive 5 разработана в 2001 году, она занимает 127,7 Мбайт памяти, что составляет 0,1% общей дисковой памяти.

Справочная система настолько сильна, что позволяет изучить систему полностью, не прибегая к сторонним источникам. К сожалению, справочная система не русифицирована, что вызывает непреодолимые препятствия для русскоязычного пользователя. Однако следует иметь в виду, что, даже хорошо зная систему Derive, можно не решить задачу, если не известны математические методы и компьютерные технологии, определяющие ее решение (в Help, как правило, они не содержатся).

## 1.4. Основы работы с Derive 5

### 1.4.1. Арифметические операторы

Арифметическими операторами системы Derive 5 являются:

- + — сложение (например,  $(a + b)$ );
- − — вычитание (например,  $(a - b)$ );
- \* — умножение (например,  $(a * b)$ );
- / — деление ( $\frac{a}{b}$ );
- ^ — возведение в степень ( $a^b$ );
- % — вычисление процентов (деление на 100 —  $\frac{a}{100}$ ).

Операция умножения при вводе математических выражений может быть реализована несколькими способами. При умножении двух чисел или двух символьных переменных необходимо применить знак умножения (\*) или нажать клавишу <Пробел>. Если же умножается число на символьную переменную, то знак умножения или <Пробел> можно не применять. При любом способе ввода знака умножения на экране символ умножения обозначается точкой.

При возведении числа или символьной переменной в отрицательную степень скобки можно не использовать. Вместо очевидной записи  $a^{-2}$  можно вводить  $a^{\wedge} - 2$ . Здесь допускаются два арифметических оператора подряд.

## 1.4.2. Алфавит системы

Алфавитом системы являются заглавные и строчные символы латинского и греческого алфавита, арабские цифры 0, 1, 2, ..., 9. При этом заглавные и строчные символы система не различает. Это ошибки грамматики, но исключительное удобство для пользователя, которому не нужно помнить, где использовать тот или иной символ. Так как клавиатура ПК не имеет греческих символов, то в Derive 5 предусмотрена панель символов, находящаяся в главном окне системы.

Специальные символы кодируются следующим образом:

- $\#e$  — основание натурального логарифма (на экране после ввода имеет вид  $e^{\wedge}$ );
- $\#i$  — мнимая единица ( $\sqrt{-1}$ ) (на экране имеет вид  $i^{\wedge}$ );
- $Pi$  — число  $\pi$ ;
- $SQRT(x)$  — корень квадратный из  $x$  (на экране имеет вид  $\sqrt{x}$ ).

Функцию  $e^x$  можно вводить как  $Exp(x)$ .

Эти и другие символы языка Derive 5 находятся на панели математических символов и символов латинского и греческого алфавитов. Имеются также некоторые символы кириллицы, например,  $\Gamma$ -символ гамма-функции.

## 1.4.3. Ввод выражений

Ввод математических выражений осуществляется в латинском алфавите. Переход в латинский алфавит осуществляется одновременным нажатием клавиш  $\langle \text{Alt} \rangle$  ( $\langle \text{Ctrl} \rangle$ ) и  $\langle \text{Shift} \rangle$  или по

щелчку мыши на кнопке **Ru** (расположенной в правом нижнем углу экрана) с последующим щелчком по тексту Английский (США).

Последовательность команд при вводе выражений:

1. **Autor Expression** (или <Ctrl>+<A>).
2. Запись выражения в диалоговом окне.
3. Нажатие клавиши <Enter> или щелчок мыши по кнопке **V** (слева от диалогового окна).

На экране появляется выражение с символом #1:, что означает номер строки, в которой оно находится. При вводе выражений можно пользоваться панелями математических символов, приведенных в двух нижних строках экрана. Это бывает полезным при вводе таких выражений, как  $\hat{e}$  или корень квадратный, которые традиционно записываются в виде  $Exp(x)$  и  $SQRT(x)$ , или при вводе выражений с символами греческого алфавита.

*Упражнение.*

Введите следующие выражения:

$$x^2 + 2x - 1;$$

$$\sin(x);$$

$$\lambda + 2\Delta = \xi;$$

$$\hat{e}^3 + \pi x.$$

При вводе сложных выражений полезно вводить их по частям, а затем собирать в общее выражение, оперируя номерами строк, в которых находятся отдельные части этого выражения. При таком вводе исчезают сложности с применением скобок и записью многоступенчатых дробей.

*Пример.*

Пусть необходимо ввести выражение:

$$\left( \frac{\sqrt{x} + 2n! - 1}{\sin(x) + \cos(x)} \right) \left( \frac{2a^2 + (3e^x - 1)x}{n \left( 1 + \frac{\sin(x)}{\cos(x)} \right)} \right) - 5.$$

Процедура ввода этого выражения:

#1:  $\sqrt{x} + 3 * n! - 1$

#2:  $\sin(x) + \cos(x)$

#3:  $2 * a^2 + (3 * e^x - 1) * x$

#4:  $n * (1 + \sin(x) / \cos(x))$

#5:  $\#1 / \#2 * \#3 / \#4 - 5$

После команды ввода (клавиша <Enter>) на экране появится исходное выражение.

Часто при решении математических задач приходится вводить выражения, в которых имеются переменные с индексами, например —  $a_1$ ,  $P_0$ ,  $K_{abc}$  и т. д. Если не настроить систему на ввод переменных с индексами, то система будет выводить выражения:  $a*1$ ,  $P*0$ ,  $K*a*b*c$ , т. е. произведения переменных и их индексов. Для настройки системы на ввод переменных с индексами необходимо исполнить следующие действия:

☐ **Declare | Input;**

☐ в появившемся окне щелкнуть мышью по кнопке **Word;**

☐ после команды ввода (клавиша <Enter>) на экране появится соответствующее сообщение о настройке системы на прием переменных с индексами.

Теперь при вводе на экране будут отображаться выражения:  $a_1$ ,  $P_0$ ,  $K_{abc}$ .

## 1.4.4. Вычисление функций

Вычисление функций в системе Derive 5 осуществляется несколькими способами.

*Способ 1.* Присваивание аргументам функции символьных или численных значений переменных с помощью символа присваивания ( $:=$ ) с последующим вычислением значения функции.

Технология этого способа состоит в выполнении пользователем следующих действий:

- ввод функции или выражения;
- присваивание аргументам численных или символьных переменных с помощью знака присваивания ( $:=$ ), например,  $x:=1.3$  или  $x:=a+e^{-b}$ ;
- выделение выражения на экране;
- команда **Simplify** или **Approximate**.

Этот способ допускает переприсваивание значений переменных. При этом прежние их значения можно не удалять с экрана (они не будут участвовать в вычислениях). Если необходимо повторить вычисления с прежними переменными, то их нужно восстановить или вновь вывести на экран. Восстановление осуществляется путем выделения переменной и нажатии клавиши  $\langle F3 \rangle$ . В строке ввода образуется переменная, которая выводится на экран нажатием клавиши  $\langle \text{Enter} \rangle$ .

*Пример 1.* Необходимо вычислить значение функции  $y = 2e^{-0,5x} + 1$  при  $x = 2,4$ .

Процедуры решения задачи:

- ввод выражения:  $y = 2e^{-0,5x} + 1$ ;
- ввод переменной с присвоением:  $x = 2,4$ ;
- выделение выражения:  $y = e^{-0,5x} + 1$ ;
- выполнение команды **Approximate** (кнопка  $\approx$ ) на панели инструментов).

На экране ответ:  $y = 1,602\dots$

*Пример 2.* Вычислить значение функции  $z = 2\ln(x + y) + \ln(x - y) + 1$  при  $x = a, y = b$ .

Процедура вычисления этой функции на экране монитора показана на рис. 1.15.

Дополнительно на рис. 1.15 приведен вариант вычисления функции из примера 2 при  $a = 4,3$ ;  $b = 2,7$  с численным ответом.

```

#1:  y = 2 · e-0.5 · x + 1
#2:  x := 2.4
#3:                                     y = 1.602388423
#4:  z = 2 · LN(x + y) + LN(x - y) + 1
#5:  x := a
#6:  y := b
#7:                                     z = LN(a - b) + 2 · LN(a + b) + 1
#8:  a := 4.3
#9:  b := 2.7
#10:                                     z = 5.361823927

```

**Рис. 1.15.** Вычисление функции из примера 2

*Способ 2.* Вычисление значения функции непосредственно по ее имени и значению аргумента.

Этот способ возможен только для случая встроенных функций: тригонометрических, алгебраических, типа  $y = \ln(x)$ ,  $y = e^x$ ,  $x!$ , и ряда специальных функций. Процедуры здесь очевидны. Вводится выражение с численным значением аргумента (например,  $y = \sin(2.1)$ ) и нажимается кнопка **Approximate** ( $\approx$ ) на панели инструментов. На экране формируется ответ:  $y=0,932\dots$ . Этот способ позволяет выполнить вычисления значений функций с наперед заданной точностью. Для установления точности, определяемой числом знаков после запятой, следует воспользоваться пунктом **Simplify** главного меню. При вызове **Simplify | Approximate** на экране появляется окно **Approximate Expression**. Желаемое число знаков ответа устанавливается в области **Digits of precision**. По умолчанию число знаков равно 10 (рис. 1.16).

После установки необходимого числа знаков делается щелчок мышью по кнопке **Approximate**, и на экране появляется ответ. Если щелкнуть по кнопке **OK**, то на экране появляется выражение: **APPROX** ( $y=\sin(1.2)$ , 4). В данном случае при вычислении значения функции  $y = \sin(1,2)$  была установлена точность, соответствующая четырем цифрам после запятой. Теперь для получения ответа необходимо щелкнуть по кнопке (=) панели инструмен-

тов. Процедуры вычисления отображаются на экране так, как это показано на рис. 1.17.

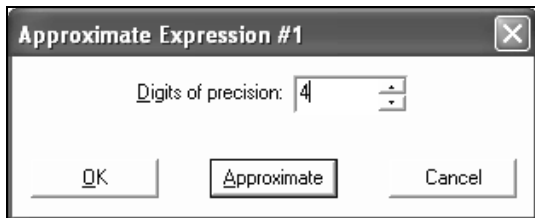


Рис. 1.16. Окно установки числа знаков вычисления функции

#1:	$y = \sin(1.2)$	
#2:		$y = 0.9320$
#3:	$\text{APPROX}(y = \sin(1.2), 4)$	
#4:		$y = 0.9320$

Рис. 1.17. Расчет функции  $y = \sin(1,2)$

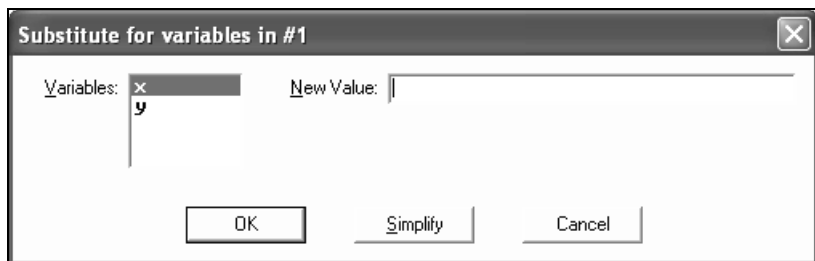
**Способ 3.** Вычисление значения функции с помощью команды **Variable Substitution**. Этот способ наиболее экономичный и универсальный. Им особенно выгодно пользоваться тогда, когда необходимо вычислять значение функции многократно при различных значениях аргумента.

Технология этого способа заключается в следующем:

- ☐ ввод функции  $y = f(x)$ , например,  $y = \cos(x)$ ;
- ☐ последовательный ввод команд **Simplify**, **Variable Substitution** или выбор кнопки **Sub** на панели инструментов (на экране формируется окно **Substitute for variables** (рис. 1.18);
- ☐ в поле ввода **Variables** указывается имя переменной  $x$ , в строке ввода **New Value** вводится численное значение этой переменной, например, 1,2 ;
- ☐ щелчок мыши по кнопке **Simplify** (на экране формируется ответ в виде  $y = \cos\left(\frac{6}{5}\right)$ );



- ❑ щелчок мыши по кнопке **Approximate** ( $\approx$ ) панели инструментов (на экране ответ:  $y=0,362\dots$ ).



**Рис. 1.18.** Окно ввода данных

Если в окне **Substitute for variables** щелкнуть по кнопке **OK**, то на экране будет выражение функции с численным значением аргумента (в нашем случае  $y = \cos(1.2)$ ). Теперь достаточно щелкнуть по кнопке **Approximate**, и на экране появится ответ:  $y=0,362\dots$ .

Процедуры вычисления на экране будут иметь вид, показанный на рис. 1.19.

#1:  $y = \cos(x)$

#2:  $y = \cos\left(\frac{6}{5}\right)$

#3:  $y = 0.3623577544$

#4:  $y = \cos(1.2)$

#5:  $y = 0.3623577544$

**Рис. 1.19.** Процедуры вычисления, реализуемые в соответствии со способом 3

*Упражнения.*

1. Вычислить выражения:

$$\frac{ax+b}{\sin(x)}, a=0,5, b=-3,25, x=7,5$$

$$\frac{an!}{a-n!}, a=2, n=5; a=2, n=10; a=2, n=20$$

$$xe^{-ax} + a, a=0,1, x=1,5; a=1, x=1; a=5, x=3$$

2. Вычислить значения  $e$  и  $\pi$  с точностью 10 и 100 знаков после запятой (установка числа знаков осуществляется с помощью выполнения цепочки команд: **Declare**, **Output Settings**, число знаков во вставке **Digits**).

### Примечание

При вычислениях пользуйтесь командами **Approximate** и **Simplify**.

## 1.4.5. Создание функции пользователя

Решение задач по специальности часто требует многократного вычисления одной и той же функции, которая не является встроенной для данной системы. Система Derive 5 позволяет создать такую функцию и обращаться к ней по имени, указывая значения аргументов, подобно встроенным функциям.

Технологию создания функций пользователя рассмотрим на примере. Предположим, что нам часто приходится вычислять функцию

$$P(t) = \frac{e^x}{x!} \text{ при разных значениях } x.$$

Создание такой функции в системе Derive 5 осуществляется следующим образом.

Активизируется пункт главного меню **Declare | Function Definition**. В появившемся окне **Declare Function Definition**, в строке ввода **Function Name and Arguments**, вводится имя функции (в нашем случае это  $P(x)$ ). В строке ввода **Function Definition** вводится функция  $e^x / x!$  (рис. 1.20).

После нажатия кнопки **ОК** на экране появляется функция  $P(x)$ . Теперь этой функцией можно пользоваться как любой другой встроенной функцией, например,  $\sin(x)$ ,  $\ln(x)$  и т. п. Для проведения вычислений необходимо ввести с клавиатуры выражение  $P(a)$ . На экране появляется функция  $P(a)$ . Затем необходимо

щелкнуть мышью по кнопке  $=$  на панели инструментов, если  $a$  — символьная переменная, и по кнопке  $\approx$ , если  $a$  — число. После этого на экране появится ответ.

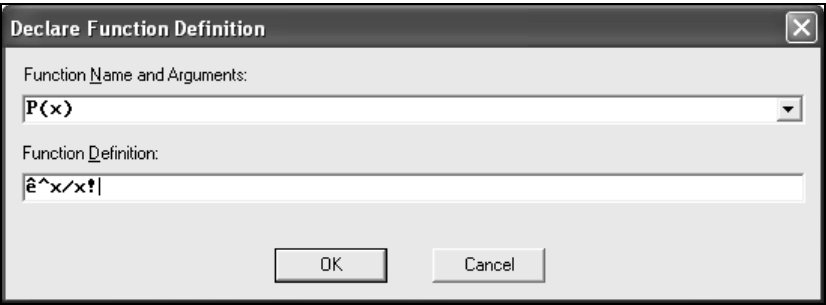


Рис. 1.20. Окно создания функции пользователя

**Примечание**

Следует иметь в виду, что аргумент  $x$  может быть также любым выражением, например —  $\sin(1) + \cos(1)$ .

В табл. 1.1 приведены результаты вычисления функции  $e^x / x!$  для различных значений  $x$ . Повторите эти вычисления.

**Таблица 1.1.** Результаты вычисления функции  $e^x / x!$

Ввод	$P(a)$	$P(3)$	$P(0)$	$P(\infty)$	$P\left(\frac{\sin(1) + \cos(1)}{2}\right)$
Ответ	$\frac{e^a}{a!}$	3,347...	1	?	2,2002

Над функцией  $P(x)$  можно выполнять любые действия: вычислять производную, интеграл, табулировать и т. п.  
После проведенных преобразований и вычислений обычно требуется произвести сохранение функции пользователя.

Для сохранения функции  $P(x)$  необходимо вызвать пункт главного меню **File | Save**. В результате этого появляется окно **Save As**. В строке ввода имени файла надо ввести желаемое имя файла (пусть это будет Pol) и нажать кнопку **Save** (Сохранить). Теперь

можно убрать с экрана исходную функцию  $P(x) = \frac{e^x}{x!}$ . Она бу-

дет сохранена, и ею можно пользоваться в данном сеансе работы системы. При переходе на новый сеанс необходимо обратиться к файлу Pol (**File | New**).

Для удаления функции  $P(x)$  достаточно удалить файл Pol обычным способом.

## Глава 2



# Визуализация вычислений

Построение графиков бывает полезно, а иногда и необходимо. Они могут строиться в следующих случаях:

- ☐ визуализации функций, являющихся моделями физических и других изучаемых явлений;
- ☐ наглядное представление пространственных фигур;
- ☐ изучение особенностей математических функций (наличие экстремумов, точек перегиба, разрывов непрерывностей и т. п.);
- ☐ выбора вида функции интерполяции;
- ☐ определения областей изоляции корней уравнения.

Система Derive является наиболее простой системой визуализации математических и физических понятий. Ее освоение не требует детального описания способов построения графиков. Для этого достаточно знать лишь назначение команд меню и особенно кнопок панели инструментов графических окон двумерной и трехмерной графики. Ниже приводится назначение команд меню и кнопок панели инструментов.

## 2.1. Двумерная графика

- ☐ **File** — просмотр графиков и их печать, выход из графического окна;
- ☐ **Edit** — создание и уничтожение аннотаций, удаление последнего графика, копирование графика в буфер обмена;

- ☐ **Insert** — открытие графических окон, аннотация графика;
- ☐ **Set** — установка масштабов и размера графика, центрирование графика;
- ☐ **Options** — установка опций;
- ☐ **Windows** — управление окнами;
- ☐ **Help** — вывод справки о Derive.

В большинстве случаев при работе с графикой нет необходимости обращаться к командам главного меню. Быстрый способ построения графиков осуществляется с помощью кнопок панели инструментов.

Исключением является пункт меню **Options** — установка опций графика, осей координат, координатной сетки, графического курсора. При обращении к пункту главного меню **Options** на экране появляется окно с командами этого меню. Щелчок по пункту **Display** вызывает появление нового окна **Display Options** (рис. 2.1).

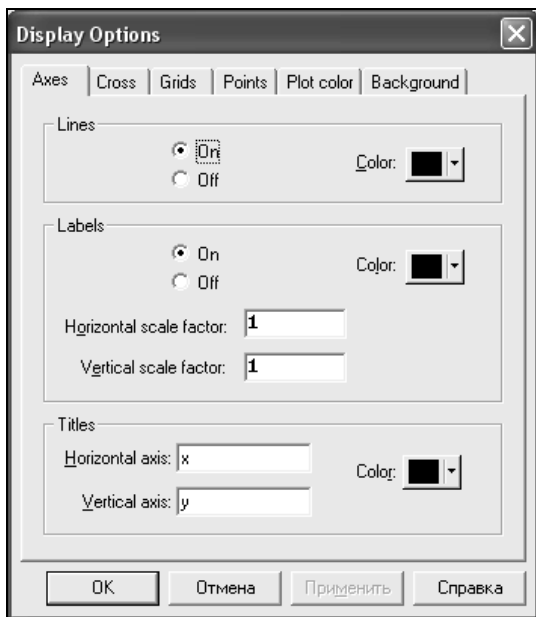


Рис. 2.1. Окно **Display Options** графического меню **Options**

Окно содержит три вкладки панели и шесть следующих команд:

- ☐ **Axes** — наличие или отсутствие осей координат (**Lines**), установка параметров координатных осей  $x$ ,  $y$  и их цвета (**Labels**), установка имени осей координат (**Titles**);
- ☐ **Cross** — установка параметров графического курсора (признак наличия и цвет);
- ☐ **Grids** — установка параметров сетки имеет две вкладки: **Display** и **Interval**, причем вкладка **Display** имеет три позиции (**Lines** — сетка линиями, **Points** — сетка точками, **Off** — линии отсутствуют), а вкладка **Interval** имеет позиции **Horizontal** и **Vertical** (в скобках отмечено общее число интервалов по осям  $x$  и  $y$ );
- ☐ **Points** — установка опций точек с двумя вкладками: **Connect** и **Size** (наличие или отсутствие кнопок и их размер);
- ☐ **Plot color** — установка цвета линий графика;
- ☐ **Background** — установка фона цвета графиков.

Цвет линий графика можно выбрать, щелкая мышью по кнопке **Plot Expression** и наблюдая за изменениями цвета графика.

Пункт меню **Insert | Annotation** (главное меню системы) позволяет установить имя рисунка. При обращении к этому пункту появляется новое окно **2D-plot Window Annotation** (рис. 2.2). В нем находится окно для текста с возможностью выбора фона и цвета (**Font, Color**). В областях ввода панели **Position** можно установить координаты текста по горизонтали (**Horizontal**) и вертикали (**Vertical**). Впрочем, текст можно переместить с помощью мыши обычным способом.

Быстрый способ построения графиков осуществляется с помощью кнопок панели инструментов. Рассмотрим их и кратко определим назначение каждой.

- ☐ Копирование и печать графика:
  - **New** — открытие нового окна системы Derive;
  - **Open** — открытие окон и демонстрация примеров решения задач;

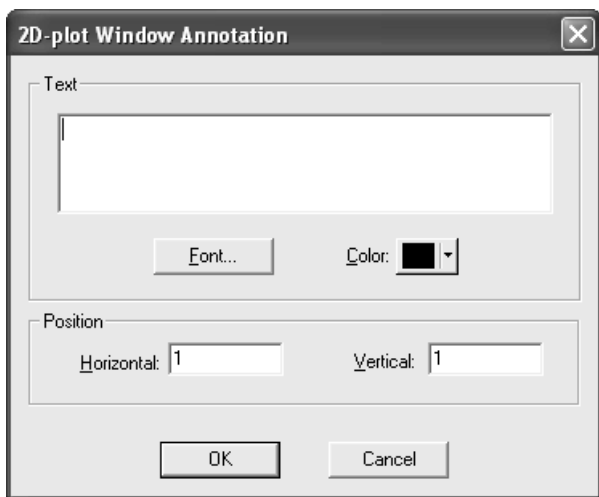


Рис. 2.2. Окно определения имени рисунка

- **Save** — сохранение графика;
  - **Print** — вывод графика на принтер;
  - **Copi Plot Window** — копирование графика в буфер обмена.
- ☐ Управление построением графика:
- **Plot expression** — построение графика выделенного выражения;
  - **Delete last plot** — удаление с экрана последнего построенного графика;
  - **Insert Annotation** — создание аннотаций.
- ☐ Центрирование и масштабирование графика:
- **Trace plots** — режим перемещения графического курсора по кривой;
  - **Center on cross** — центрирование графика относительно курсора;
  - **Center on origin** — центрирование графика по самому себе;
  - **Set range with box** — развертывание выделенного графика во все окно.



□ **Расширение области графика:**

- **Zoom out** (<F10>) — расширение графика по горизонтали и вертикали;
- **Zoom vertical out** (<F8>) — расширение графика по вертикали;
- **Zoom horizontal out** (<F6>) — расширение графика по горизонтали.

□ **Сжатие области графика:**

- **Zoom in** (<F9>) — сжатие области графика в обе стороны;
- **Zoom vertical in** (<F7>) — сжатие области графика по вертикали;
- **Zoom horizontal in** (<F5>) — сжатие области графика по горизонтали.

□ **Algebra window** (<Ctrl>+<1>) — возврат в окно выражений.

Графическое изображение кнопок на экране столь очевидно, что нет необходимости в их подробном описании. Вполне достаточно приведенного выше списка.

Процедуры графического представления функции в Derive можно определить следующим составом:

- *ввод функции  $f(x)$*  и ее выделение на экране монитора;
- *формирование окна двумерной графики* с сеткой координат — щелкнуть мышью по кнопке **2D-plot window** на панели инструментов;
- *формирование графика* выделенной функции — щелкнуть мышью по кнопке **Plot Expression** на панели инструментов.

В ряде случаев Derive устанавливает масштабы графика автоматически. Однако часто масштабы приходится устанавливать пользователю с помощью кнопок расширения и сжатия графика.

Возврат в окно выражений осуществляется с помощью нажатия кнопки **Algebra window** панели инструментов. При этом предыдущий график сохраняется, а если необходимо, то, повторяя выше описанные процедуры, можно построить в том же окне второй

график. Удаление первого графика осуществляется нажатием кнопки **Delete Last Plot**.

Рассмотрим пример. Пусть необходимо построить график функции  $y(x) = \sin(x)/x$  и выделить его центральную часть, находящуюся в диапазоне значений  $x$  от  $-3$  до  $+3$ .

Решение заключается в выполнении следующих действий:

- ☐ ввод функции  $y(x)$ ;
- ☐ формирование на экране окна двумерной графики с сеткой координат (щелкнуть мышью по кнопке **2D-plot window** на панели инструментов);
- ☐ формирование на экране (рис. 2.3) графика функции  $y(x) = \sin(x)/x$  (щелкнуть мышью по кнопке **Plot Expression** на панели инструментов);
- ☐ формирование курсора в виде креста (щелчок мышью по кнопке **Set range with box**);

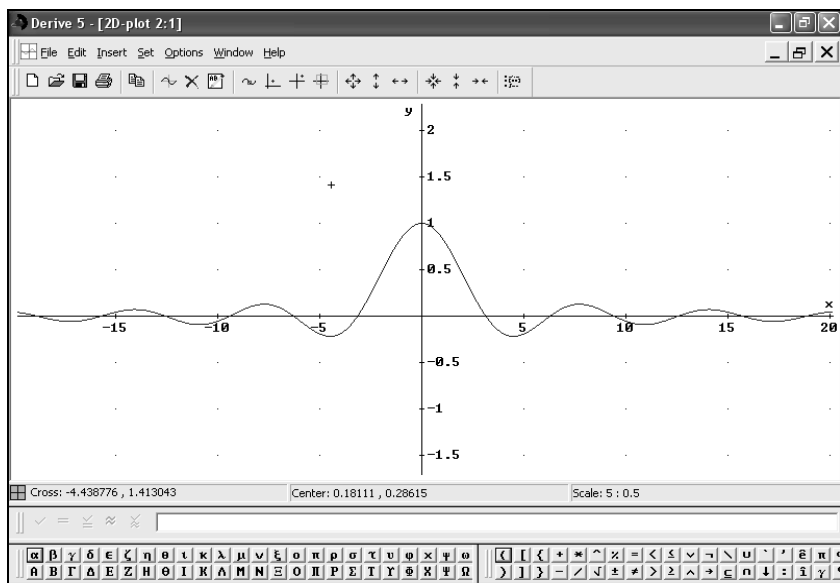


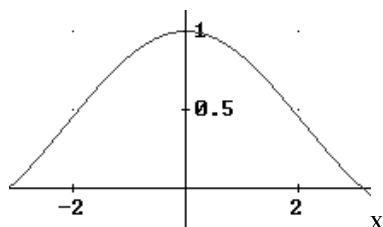
Рис. 2.3. График функции  $y(x) = \sin(x)/x$

- удерживая кнопку мыши нажатой, перемещение курсора по экрану (на экране появляется прямоугольник, который при перемещении курсора расширяется, охватывая нужную область графика);
- отпустив кнопку мыши, вызываем появление нового окна **Set 2D-Plot Range** с панелями установки диапазона изменения графика по осям  $x$  и  $y$ ;

### Примечание

Полезно в этом случае отредактировать диапазон изменения переменных (**Minimum**, **Maximum**) по оси  $x$  (**Horizontal**) и по оси  $y$  (**Vertical**), а также цену деления (**Intervals**), если это необходимо.

- по нажатию кнопки **ОК**, формирование на экране требуемого графика (рис. 2.4).



**Рис. 2.4.** Выделенная часть графика функции  $y(x) = \sin(x)/x$

### Примечание

Возвращение графика в исходное состояние осуществляется с помощью кнопок сжатия и расширения графика.

Система Derive 5 также строит графики для параметрически заданных функций и функций в полярных системах координат.

Для построения графиков параметрически заданных функций необходимо эти функции записать в виде вектора, отделив друг от друга запятой. Векторное задание функций осуществляется представлением их в квадратных скобках, например,  $[\sin(x), x * \sin(x)]$ .

Технологию построения графика параметрически заданных функций рассмотрим на примере.

- ☐ Ввод функции в виде вектора  $[f(x), v(x)]$ , например,  $[\cos(x)+1.5, \sin(x)*x]$ .
- ☐ Формирование на экране графического окна с сеткой координат (щелкнуть мышью по кнопке **2D-plot window** на панели инструментов).
- ☐ Формирование окна (рис. 2.5) **Parametric Plot Parameters** (щелкнуть мышью по кнопке **Plot Expression** на панели инструментов).
- ☐ Уточнение (при необходимости) параметров построения графика.

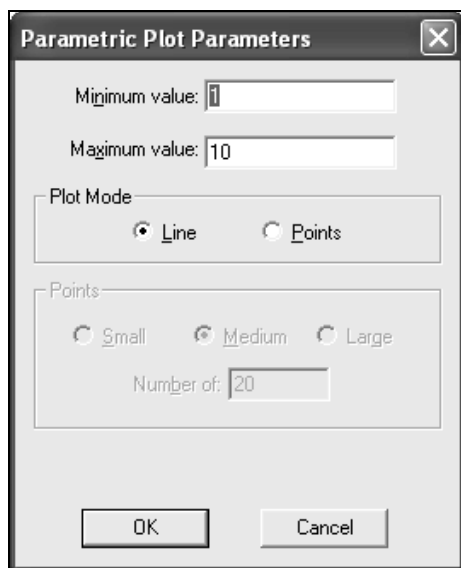
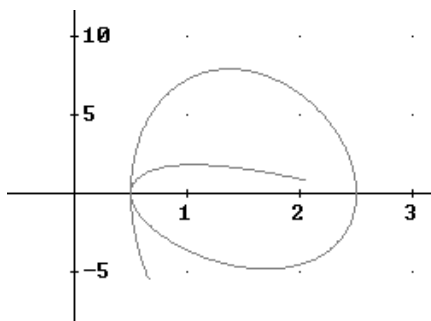


Рис. 2.5. Окно установки параметров рисунка

### Примечание

Установим диапазон изменения аргумента  $x$  от  $x_{\min} = 1$  (**Minimum value**) до  $x_{\max} = 10$  (**Maximum value**) и форму представления: линиями (**Line**) или точками (**Points**).

- Формирование на экране графика (рис. 2.6) функции (щелчок мыши на кнопке **OK**), который получен после корректировки цены деления по осям  $x$  и  $y$ . Щелкая по кнопке **Plot Expression** можно выбрать цвет графика.



**Рис. 2.6.** График параметрически заданных функций

При построении графика в полярной системе координат необходимо выполнить следующие процедуры:

- ввести функцию  $r(\theta)$ ;
- открыть окно двумерной графики (кнопка **2D-plot window**);
- установить на панели **Coordinate System** команды главного меню **Set** полярную систему координат (флажок **Polar**) и нажать кнопку **OK** (рис. 2.7);
- щелкнуть мышью по кнопке **Plot Expression** на панели инструментов (появляется окно **Parametric Plot Parameters**);
- уточнить (при необходимости) параметры построения графика и нажать кнопку **OK** (на экране будет сформирован график в полярной системе координат).

На рис. 2.8 показан график функции  $y(x) = \sin((2\theta + 1)/5, 5)$ , построенный по приведенной выше технологии.

В процессе работы с графикой, эффективным средством анализа является графический курсор. Он позволяет:

- определить координаты точек графика (режим — **Autoscale New Plots**);

- ☐ посмотреть график за пределами видимого окна путем перемещения курсора в нужную сторону (режим — **Follow Cross**);
- ☐ повысить точность определения координат графика (режим — **Trace Plots**).

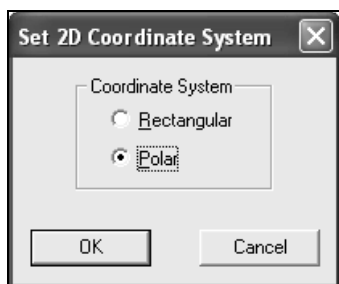


Рис. 2.7. Окно установки полярной системы координат

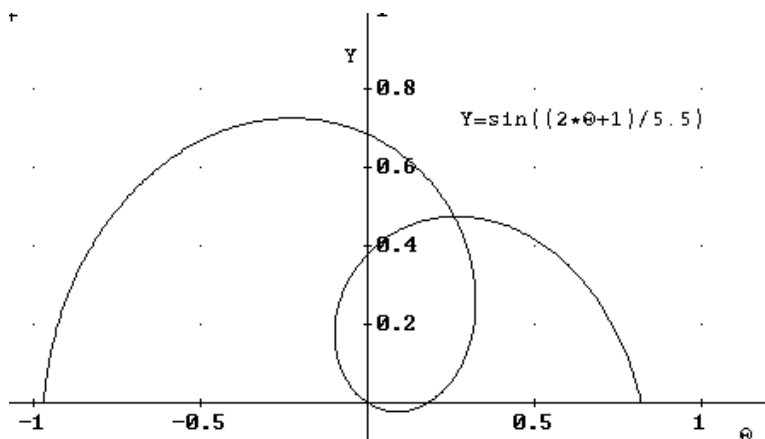


Рис. 2.8. График функции в полярной системе координат

### Примечание

В последнем режиме курсор превращается из крестика в маленький квадратик, устанавливается на необходимую кривую и перемещается мышью или клавишами точно по кривой.

Все перечисленные выше режимы действия графического курсора устанавливаются по команде **Options** графического меню.

## 2.2. Трехмерная графика

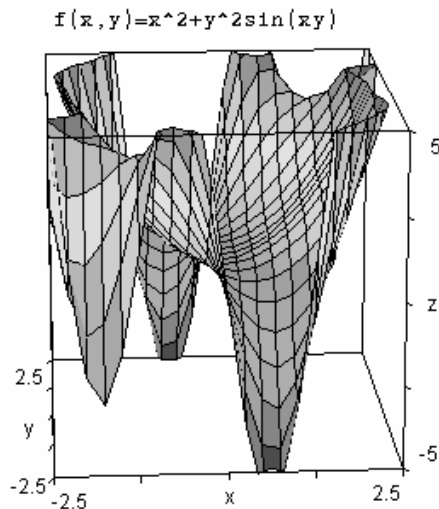
Команды главного меню трехмерной графики не отличаются от команд меню двумерной графики. Панели инструментов группы копирования и печати также идентичны. Остальные кнопки панели инструментов трехмерной графики имеют следующие значения:

- ☐ **Delete Plot** — удаление графика;
- ☐ **Plot** — построение пространственного графика выделенного выражения;
- ☐ **Insert Annotation** — создание аннотаций;
- ☐ **Trace Plots** — трассирование графика;
- ☐ **Set Plot Range** — установка масштабов по осям координат;
- ☐ **Set Eye Position** — установка размеров области графика;
- ☐ **Zoom out** (<F10>) — расширение графика в обе стороны;
- ☐ **Zoom in** (<F9>) — сжатие графика в обе стороны;
- ☐ **Rotate Plots** — вращение графика;
- ☐ **Rotate left** — поворот графика влево на одну позицию;
- ☐ **Rotate right** — поворот графика вправо на одну позицию;
- ☐ **Rotate up** — поворот графика в сторону от пользователя;
- ☐ **Rotate down** — поворот графика в сторону пользователя;
- ☐ **Magnify plot** — расширение области графика;
- ☐ **Shrinkt plot** — сжатие области графика;
- ☐ **Algebra window** — возврат в окно выражений.

Процедуры построения пространственных графиков достаточно просты и мало чем отличаются от процедур построения двумерных графиков. Кнопки панели инструментов, при подведении

к ним курсора мыши, оживляются и появляется их название. Форма кнопок является превосходной подсказкой для пользователя. Поэтому нет необходимости детально описывать процедуры построения трехмерных графиков. Пользователь легко в них разберется уже при первых попытках их построения.

На рис. 2.9 показан график функции  $f(x, y) = x^2 + y^2 \sin(xy)$ , полученный по описанной выше технологии двумерных графиков.



**Рис. 2.9.** График функции  $f(x, y) = x^2 + y^2 \sin(xy)$





# Вычисление математических функций

Решение задач с помощью Derive 5 или любого другого универсального программного средства символьной математики осуществляется с помощью функций системы. Например, алгебраические и трансцендентные уравнения решаются с помощью функции `SOLVE`, интегралы вычисляются с помощью функции `INTEGRATE`, определение суммы ряда чисел осуществляется с помощью функции `SUM` и т. д. Эти и многие другие функции будут в дальнейшем использоваться в соответствующих главах книги. В этой главе излагаются способы вычисления математических функций, которые могут вычисляться как с помощью функций системы Derive 5, так и путем прямых вычислений. Таким образом, не следует отождествлять понятия "функции системы Derive 5" и "математические функции".

Математические функции могут классифицироваться как — элементарные, специальные и функции пользователя. Ниже приводятся способы их представления и вычисления.

## 3.1. Элементарные функции

### 3.1.1. Тригонометрические функции

Эти функции вычисляются с помощью функций системы Derive 5, которые имеют следующий вид:

- $\text{SIN}(x)$  — синус угла  $x$ ;
- $\text{COS}(x)$  — косинус угла  $x$ ;

- $\text{TAN}(x)$  — тангенс угла  $x$ ;
- $\text{COT}(x)$  — котангенс угла  $x$ ;
- $\text{SEC}(x)$  — секанс угла  $x$ ;
- $\text{CSC}(x)$  — косеканс угла  $x$ .

Особенностями представления и вычисления тригонометрических функций являются:

- аргумент  $x$  записывается в круглых скобках;
- аргумент  $x$  задается в радианах.

Набор имени любой функции можно выполнять строчными или прописными символами латинского алфавита. После ввода функция на экране будет представлена всегда большими символами.

Аргумент  $x$  может быть числом вещественным или комплексным.

Система Derive 5 допускает задание  $x$  в градусах. Для этого необходимо представить аргумент в скобках в следующем виде —  $x \text{ deg}$ , где  $x$  — угол в градусах. Константа  $\text{deg}$  осуществляет пере-

вод угла из радиан в градусы и имеет значение:  $\text{deg} = \frac{180}{\pi}$ . При

таком представлении аргумента функция, например  $\sin(x)$ , на экране будет иметь вид  $\text{SIN}(x \cdot ^\circ)$ . Обозначение  $(x \cdot ^\circ)$  вместо  $\text{deg}$  можно записать, воспользовавшись таблицей символов (щелкнуть по этому знаку мышью, предварительно установив курсор в нужном месте выражения тригонометрической функции).

### 3.1.2. Обратные тригонометрические функции

Обратные тригонометрические функции вычисляют значение угла, выраженного в радианах. Они представляются в виде:

- $\text{ASIN}(x)$  — арксинус угла  $x$ ;
- $\text{ACOS}(x)$  — арккосинус угла  $x$ ;
- $\text{ATAN}(x)$  — арктангенс угла  $x$ ;
- $\text{ACOT}(x)$  — арккотангенс угла  $x$ ;

□  $ASEC(x)$  — арксеканс угла  $x$ ;

□  $ACSC(x)$  — арккосеканс угла  $x$ .

В процессе вычислений могут быть полезными следующие соотношения:

$$\square ACOS(x) = \frac{\pi}{2} - ASIN(x);$$

$$\square ACOT(x) = \frac{\pi}{2} - ATAN(x);$$

$$\square ASEC(x) = ACOS\left(\frac{1}{x}\right);$$

$$\square ACSC(x) = ASIN\left(\frac{1}{x}\right).$$

Если необходимо получить значение угла в градусах, то достаточно полученный результат разделить на константу  $deg = \frac{180}{\pi}$ .

Набор обратных тригонометрических функций можно осуществлять прописными или строчными символами или любой их комбинацией. На экране функция всегда будет представлена большими символами.

### 3.1.3. Гиперболические функции

Гиперболические функции вычисляются в Derive 5 с помощью следующих системных функций:

□  $SINH(x)$  — гиперболический синус угла  $x$ ;

□  $COSH(x)$  — гиперболический косинус угла  $x$ ;

□  $TANH(x)$  — гиперболический тангенс угла  $x$ ;

□  $COTH(x)$  — гиперболический котангенс угла  $x$ ;

□  $SECH(x)$  — гиперболический секанс угла  $x$ ;

□  $CSCH(x)$  — гиперболический косеканс угла  $x$ .

Гиперболические функции представляются выражениями, содержащими экспоненциальные функции. Запоминать их нет необходимости. Достаточно выделить функцию на экране и щелкнуть мышью по кнопке **Simplify** на панели инструментов. В результате появится гиперболическая функция, представленная через функции экспоненциальные. Эти процедуры показаны на рис. 3.1.

#1:  $\text{SINH}(x)$

$$\frac{e^x}{2} - \frac{e^{-x}}{2}$$

#2:

#3:  $\text{COSH}(x)$

$$\frac{e^x}{2} + \frac{e^{-x}}{2}$$

#4:

#5:  $\text{TANH}(x)$

$$\frac{e^{2 \cdot x} - 1}{e^{2 \cdot x} + 1}$$

#6:

#7:  $\text{COTH}(x)$

$$\frac{e^{2 \cdot x} + 1}{e^{2 \cdot x} - 1}$$

#8:

#9:  $\text{SECH}(x)$

$$\frac{2 \cdot e^x}{e^{2 \cdot x} + 1}$$

#10:

#11:  $\text{CSCH}(x)$

$$\frac{2 \cdot e^x}{e^{2 \cdot x} - 1}$$

#12:

**Рис. 3.1.** Представление гиперболических функций через экспоненциальные

### 3.1.4. Обратные гиперболические функции

Обратные гиперболические функции в системе Derive 5 вычисляются с помощью следующих системных функций:

- $\text{ASINH}(x)$  — обратный гиперболический синус угла  $x$ ;
- $\text{ACOSH}(x)$  — обратный гиперболический косинус угла  $x$ ;
- $\text{ATANH}(x)$  — обратный гиперболический тангенс угла  $x$ ;
- $\text{ACOTH}(x)$  — обратный гиперболический котангенс угла  $x$ ;
- $\text{ASECH}(x)$  — обратный гиперболический секанс угла  $x$ ;
- $\text{ACSCH}(x)$  — обратный гиперболический косеканс угла  $x$ .

Обратные гиперболические функции представляются логарифмическими. Для их получения необходимо выделить функцию на экране и щелкнуть по кнопке **Simplify** на панели инструментов. На рис. 3.2 показаны результаты таких преобразований.

#1:	$\text{ASINH}(x)$	
#2:		$\text{LN}(\sqrt{x^2 + 1} + x)$
#3:	$\text{ACOSH}(x)$	
#4:		$\hat{1} \cdot \text{ACOS}(x)$
#5:	$\text{ATANH}(x)$	
#6:		$- \hat{1} \cdot \text{ATAN}(\hat{1} \cdot x)$
#7:	$\text{ACOTH}(x)$	
#8:		$- \hat{1} \cdot \text{ACOT}(- \hat{1} \cdot x)$
#9:	$\text{ASECH}(x)$	
#10:		$\hat{1} \cdot \text{ACOS}\left(\frac{1}{x}\right)$
#11:	$\text{ACSCH}(x)$	
#12:		$\text{LN}\left[\frac{\sqrt{x^2 + 1} \cdot \text{SIGN}(x) + 1}{x}\right]$

**Рис. 3.2.** Представление обратных гиперболических функций логарифмическими

Обратные гиперболические функции  $\text{ACOSH}(x)$ ,  $\text{ATANH}(x)$ ,  $\text{ACOTH}(x)$  представляются через логарифмические следующим образом:

$$\square \text{ACOSH}(x) = \text{Ln}(x + \sqrt{x^2 - 1}) = i \times \text{ACOS}(x);$$

$$\square \text{ATANH}(x) = \frac{1}{2} \text{Ln}((1+x)/(1-x)) = i \times \text{ATAN}(ix);$$

$$\square \text{ACOTH}(x) = \frac{1}{2} \text{Ln}((1+x)/(1-x)) = i \times \text{ATAN}(-ix).$$

### 3.1.5. Логарифмическая функция

Система Derive 5 имеет две функции вычисления логарифма числа  $x$ :

$\square \text{LN}(x)$  — значение натурального логарифма (логарифма с основанием  $e$ ) числа  $x$ ;

$\square \text{LOG}(x, z)$  — значение логарифма числа  $x$  по основанию  $z$ .

Функции для вычисления антилогарифма числа  $x$  в Derive 5 нет.

Эту функцию можно вычислить по соотношению  $x = z^k$ , где  $k$  — значение логарифма числа  $x$  при основании  $z$ .

### 3.1.6. Экспоненциальная функция

Функцию  $e^x$  можно вычислить следующими тремя способами:

$\square$  с помощью системной функции  $\text{EXP}(x)$ ;

$\square$  путем представления функции в виде степени  $e$  ( $\#e \wedge x$ );

$\square$  путем представления функции в виде степени  $e^{\wedge} \wedge x$  с использованием символа  $e^{\wedge}$  на панели специальных символов.

### 3.1.7. Функция извлечения корня квадратного из числа $x$

Извлечение квадратного корня числа  $x$  можно осуществить следующими тремя способами:

- ☐ с помощью системной функции `SQRT(x)`;
- ☐ путем вычисления выражения  $x \wedge 0,5$ ;
- ☐ с помощью знака  $\sqrt{\quad}$  панели инструментов.

## 3.2. Специальные функции

Derive 5 имеет сотни специальных функций.

#### Примечание

В прежних версиях системы многие из этих функций находились в библиотечных файлах. Перед использованием их нужно было загружать в память компьютера, используя меню системы (**File | Load | Vtility**) с последующим заданием имени файла. В Derive 5 такая загрузка не всегда требуется.

Описанные выше способы вычисления элементарных функций применимы и для вычисления специальных функций. При этом основным из них является способ, при котором используются системные функции Derive 5. При обращении к ним, в случае задания символьных переменных, на экране появляются расчетные формулы. Наличие расчетных формул позволяет проверять правильность вычисления специальной функции. Для этого достаточно выделить формулу, подставить численные значения переменным с помощью команды **SUB** и щелкнуть мышью по кнопке **Approximate** на панели инструментов. Пользователю не нужно помнить аналитические выражения специальных функций и способы их представления при численных расчетах (с помощью рядов, интегральных соотношений, элементарных функций и др.).

Наиболее экономичной является следующая технология вычисления специальных функций:

- ввод функции в общем виде (без подстановки численных значений аргументов);
- подстановка численных значений аргументов с помощью команды **Sub** (определена соответствующей кнопкой на панели инструментов);
- выполнение команды **Approximate**.

Эта технология не требует редактирования функции при повторных ее вычислениях с новыми значениями аргументов. Кроме того, она сохраняет выражение функции на экране, которое можно всегда вызвать в окно ввода выражений нажатием клавиши <F3>.

Ниже приводятся специальные функции, которые часто используются в научных и инженерных расчетах. Состав встроенных специальных функций системы Derive 5 огромный. С ним можно ознакомиться и изучить, обратившись к пункту помощи **Help** главного меню системы.

### 3.2.1. Математические функции

Здесь не будут рассматриваться математические функции, которые называются часто элементарными и которые рассмотрены в *разд. 3.1*. Основное внимание сейчас будет сосредоточено на функциях комплексного аргумента, комбинаторики, генерации случайных чисел и некоторых числовых функциях.

#### Функции комплексного аргумента

$ABS(Z)$  — модуль комплексного числа  $z$ ;

$RE(Z)$  — действительная часть комплексного числа;

$IM(Z)$  — мнимая часть комплексного числа;

$CONJ(Z)$  — комплексно-сопряженное число;

$PHASE(Z)$  — фаза комплексно-сопряженного числа.



*Пример 3.1.* Пусть имеется комплексное число  $z = 1,5 - 0,5i$ , которое является аргументом функции  $\cos(z)$ . Требуется определить все приведенные выше функции косинуса комплексного числа  $z$ .

Вычислительные процедуры и ответы приведены на рис. 3.3.

```
#1: 1.5 - 0.5·î
#2: COS(1.5 - 0.5·î)
#3: 0.07976510530 + 0.5197899547·î
#4: |0.07976510530 + 0.5197899547·î|
#5: 0.5258745753
#6: RE(0.07976510530 + 0.5197899547·î)
#7: 0.07976510529
#8: IM(0.07976510530 + 0.5197899547·î)
#9: 0.5197899546
#10: CONJ(0.07976510530 + 0.5197899547·î)
#11: 0.07976510529 - 0.5197899546·î
#12: PHASE(0.07976510530 + 0.5197899547·î)
#13: 1.418527743
```

**Рис. 3.3.** Вычислительные процедуры CONJ и PHASE функции комплексного числа

## Функции комбинаторики

К комбинаторным функциям системы Derive 5 относятся функции вычисления числа сочетаний  $C_n^m$  и размещений  $P_n^m$ . Эти функции имеют вид:

$$C_n^m = \frac{n!}{(n-m)!m!},$$

$$P_n^m = \frac{n!}{(n-m)!}.$$

Все комбинаторные функции вычисляются по значениям факториалов числа  $n$  и  $m$ , поэтому гамма-функцию и функцию  $z!$  также относят к комбинаторным функциям. Так как эти функции имеют широкое применение в математических расчетах и применимы для действительных и комплексных, целых и дробных, положительных и отрицательных чисел, то они будут рассмотрены отдельно.

Функциями комбинаторики в Derive 5 являются:

□  $\text{COMB}(n, m)$  — число сочетаний из  $n$  элементов по  $m$ ;

□  $\text{PERM}(n, m)$  — число размещений из  $n$  элементов по  $m$ .

Вычисления комбинаторных функций приведены ниже (рис. 3.4) при  $n = 5$  и  $m = 3$ . Технология вычислений такая же, как и в случае вычисления функции комплексного аргумента.

#1:	$\text{COMB}(n, m)$	
#2:		$\frac{n!}{m! \cdot (n - m)!}$
#3:	$\text{PERM}(n, m)$	
#4:		$\frac{n!}{(n - m)!}$
#5:	$\text{COMB}(5, 3)$	
#6:		10
#7:	$\text{PERM}(5, 3)$	
#8:		60

Рис. 3.4. Определение значений комбинаторных функций

В строках #2 и #4 (рис. 3.6) приведены математические выражения числа сочетаний и размещений из  $n$  элементов по  $m$ . Формулы являются откликами на функции  $\text{COMB}(n, m)$  и  $\text{PERM}(n, m)$ , полученными при нажатии кнопки **Simplify** панели инструментов. В строках #6 и #8 находятся численные значения функций при  $n = 5$  и  $m = 3$ .

## Функция генерации случайных чисел

При статистическом моделировании случайных процессов необходимо иметь случайные числа, полученные при определенном законе распределения случайной величины. Генерация случайных чисел в Derive 5 осуществляется с помощью функции `RANDOM(n)`, которая возвращает случайное число из диапазона  $[0..n]$  при равномерном распределении случайных величин. Компьютерной технологией генерации случайных чисел могут быть следующие два способа.

1. Набрать функцию, например `RANDOM(10)`, в окне ввода и, не выводя ее на экран, щелкнуть по кнопке **Approximate** ( $\approx$ ), расположенной слева от окна ввода. На экране после каждого щелчка будет появляться случайное число из диапазона чисел  $[0..9]$ . Недостаток этого способа заключается в том, что числа на экране располагаются в столбик и не образуют вектора случайных чисел, который необходим для дальнейших расчетов.
2. Образование вектора случайных чисел по программе, которая имеет следующий вид: `VECTOR(RANDOM(n), k, 1, m)`, где  $n$  — диапазон случайных чисел,  $k$  — последовательность образования случайных чисел,  $m$  — количество случайных чисел.

Образование вектора случайных чисел для второго случая, при  $n = 1, k = 1, m = 15$ , показано на рис. 3.5.

```
#1:  RANDOM(n)
#2:  VECTOR(RANDOM(n), k, 1, m)
#3:  VECTOR(RANDOM(1), k, 1, 15)
#4:  [0.3397745829, 0.2310834777, 0.9311302541, 0.5744827687, 0.9085190132, 0.8353277490,
      0.7978474231, 0.9243058653, 0.9700232673, 0.2206409413, 0.6458383624, 0.7176009900,
      0.4321013710, 0.1992560261, 0.3890206136]
```

Рис. 3.5. Образование вектора случайных чисел

## Числовые функции

В данном разделе рассматриваются функции, аргументы которых являются действительными числами.

$\text{ABS}(x)$  — определение абсолютного значения  $x$  (на экране представляется в виде  $|x|$ ):

- $\text{MAX}(a, b, \dots)$  — возвращает максимальное значение вектора чисел;
- $\text{MIN}(a, b, \dots)$  — возвращает минимальное значение вектора чисел;
- $\text{GCD}(m, n, \dots)$  — вычисляет наибольший общий делитель чисел  $m, n, \dots$ ;
- $\text{LCM}(m, n, \dots)$  — возвращает наименьшее общее кратное чисел  $m, n, \dots$  (общий знаменатель дробных чисел);
- $\text{FLOOR}(m, n)$  — возвращает наибольшее целое число, которое меньше или равно отношению чисел  $m$  и  $n$ ;
- $\text{FLOOR}(m)$  — возвращает целую часть числа  $m$ ;
- $\text{MOD}(m, n)$  — возвращает положительный остаток от деления  $m$  на  $n$  ( $m$  по модулю  $n$ );
- $\text{MOD}(m)$  — возвращает дробную часть числа  $m$ ;
- $\text{SIGN}(x)$  — возвращает 1, если  $x > 0$ , -1, если  $x < 0$ , и  $\pm 1$  при  $x = 0$ .

Реализация этих функций особенностей не имеет. Пример расчета показан на рис. 3.6.

```
#1:  |-3.25|
#2:                                     3.25
#3:  [1, 3.2, 5.1, -1.5, 5.8, 6.1, 0.65, 7, 6.2, 4.1]
#4:  MAX([1, 3.2, 5.1, -1.5, 5.8, 6.1, 0.65, 7, 6.2, 4.1])
#5:                                     7
#6:  MIN([1, 3.2, 5.1, -1.5, 5.8, 6.1, 0.65, 7, 6.2, 4.1])
#7:                                     -1.5
#8:                                     [1, 16/5, 51/10, -3/2, 29/5, 61/10, 13/20, 7, 31/5, 41/10]
#9:  GCD(14, 35, 49, 56, 77, 105, 147, 210)
#10:                                     7
#11:  LCM(3, 5, 8, 12, 26, 34)
#12:                                     26520
#13:  FLOOR(21, 6)
#14:                                     3
```

**Рис. 3.6.** Расчет числовых функций (начало)

#15:	FLOOR(4.23)	
#16:		4
#17:	MOD(34, 7)	
#18:		6
#19:	MOD(3.47)	
#20:		0.47
#21:	SIGN(-3.54)	
#22:		-1

Рис. 3.6 (окончание)

## Статистические функции и функции ошибок

Основными статистическими функциями Derive 5 являются:

- AVERAGE(V) — среднее арифметическое значение случайных величин, представленных в виде вектора V;
- RMS(V) — среднеквадратическое значение случайных величин;
- VARIACE(V) — дисперсия случайных величин;
- STDEV(V) — среднеквадратичное отклонение;
- NORMAL(x, m, d) — нормальное распределение случайной величины x, имеющей среднее значение, равное m, и среднеквадратическое отклонение d;
- NORMAL(Z) — интегральное распределение случайной величины;
- ERF(x) — функция ошибок;
- ERF(x, y) — функция ошибок общего вида;
- ERFC(x) — дополнительная функция ошибок.

Реализацию статистических функций в Derive 5 рассмотрим на примерах.

*Пример 3.2.* Получить статистические функции для случая символьных переменных a, b, c.

Решение задачи представлено на рис. 3.7.

#1: AVERAGE(a, b, c)

#2: 
$$\frac{a + b + c}{3}$$

#3: RMS(a, b, c)

#4: 
$$\frac{\sqrt{3 \cdot \sqrt{a^2 + b^2 + c^2}}}{3}$$

#5: VARIANCE(a, b)

#6: 
$$\frac{(a - b)^2}{2}$$

#7: STDEV(a, b)

#8: 
$$\frac{\sqrt{2 \cdot |a - b|}}{2}$$

**Рис. 3.7.** Получение статистических функций AVERAGE, RMS, VARIANCE, STDEV

*Пример 3.3.* Пусть необходимо образовать, с помощью функции RANDOM(n), вектор случайных величин, состоящий из 20 чисел в диапазоне от 1 до 10, а также вычислить основные статистические характеристики: среднее значение, среднеквадратическое значение, дисперсию и среднеквадратическое отклонение.

Решение задачи заключается в следующем.

Вектор случайных чисел образуем с помощью программы VECTOR(RANDOM(10),k,1,20), которая была рассмотрена ранее.

Дальнейшая технология решения задачи состоит в следующем:

- ❑ ввод функции AVERAGE(#4), где #4 — номер строки, в которой находится вектор v случайных величин (на экране образуется функция AVERAGE(v), v — вектор случайных чисел);
- ❑ нажатие кнопки **Approximate** на панели инструментов (на экране формируется среднеарифметическое значение случайных величин);
- ❑ технология повторяется для вычисления остальных характеристик вектора случайных чисел.

```

#1:  RANDOM(10)
#2:  VECTOR(RANDOM(10), k, 1, 20)
#3:  [3, 5, 2, 5, 9, 9, 5, 9, 9, 5, 8, 4, 7, 5, 9, 0, 9, 6, 2, 0]
#4:  AVERAGE([3, 5, 2, 5, 9, 9, 5, 9, 9, 5, 8, 4, 7, 5, 9, 0, 9, 6, 2, 0])
#5:  5.55
#6:  RMS([3, 5, 2, 5, 9, 9, 5, 9, 9, 5, 8, 4, 7, 5, 9, 0, 9, 6, 2, 0])
#7:  6.296824596
#8:  VARIANCE([3, 5, 2, 5, 9, 9, 5, 9, 9, 5, 8, 4, 7, 5, 9, 0, 9, 6, 2, 0])
#9:  9.313157894
#10:  STDEV([3, 5, 2, 5, 9, 9, 5, 9, 9, 5, 8, 4, 7, 5, 9, 0, 9, 6, 2, 0])
#11:  3.051746695

```

Рис. 3.8. Формирование вектора случайных чисел

```

#1:  NORMAL(x, m, d)
#2:  
$$\frac{\text{ERF}\left(\frac{\sqrt{2} \cdot (x - m)}{2 \cdot d}\right) + 1}{2}$$

#3:  NORMAL(x)
#4:  
$$\frac{\text{ERF}\left(\frac{\sqrt{2} \cdot x}{2}\right) + 1}{2}$$

#5:  NORMAL(2, 1.5, 1)
#6:  0.6914624612
#7:  NORMAL(2)
#8:  0.9772498680
#9:  ERF(x, y)
#10:  ERF(y) - ERF(x)
#11:  ERFC(x)
#12:  1 - ERF(x)
#13:  ERF(0.5)
#14:  0.5204998778
#15:  ERF(0.5, 2.5)
#16:  0.4790931701
#17:  ERFC(0.5, 2.5)
#18:  0.4795001221

```

Рис. 3.9. Вычисление статистических функций и функции ошибок при символьных и численных значениях аргументов

Процедуры решения задачи и конечные результаты приведены на рис. 3.8.

*Пример 3.4.* Вычислить статистические функции и функции ошибок при символьных и численных значениях аргументов.

Вычислительные процедуры и результаты расчетов на экране монитора имеют вид, показанный на рис. 3.9.

На рис. 3.9, в строках #1-#4 и #9-#12, приведены значения функций при задании аргументов в символьном виде. В остальных строках показаны результаты решения нашего примера.

Следует иметь в виду, что в прежних версиях Derive функции VARIANCE нет. Вместо нее имеется функция VAR(V), однако отклики этих функций разные.

## Вероятностные функции

Кроме рассмотренных выше, в файле PROBABIL.MTH имеются вероятностные функции, которые часто используют в вероятностных и статистических расчетах.

□  $\text{PSI}(x)$  — пси-функция Эйлера. По определению она представляется в следующем виде:

$$\psi(x) = \frac{d}{dx} \ln \Gamma(x);$$

и может иметь множество представлений: интегральных, в виде ряда, в виде произведения. Наиболее простым из них является следующее:

$$\psi(x) = \frac{d}{dx} \ln \Gamma(x) = \int_0^{\infty} \left( \frac{e^{-t}}{t} - \frac{e^{-xt}}{1-e^{-t}} \right) dt.$$

В системе Derive 5 для определения пси-функции используется более сложное интегральное представление:

- $\text{POISSON\_DENSITY}(k, t)$  — плотность вероятности распределения Пуассона, имеющее вид  $-\frac{e^{-t} t^k}{k!}$ ;



- `POISSON_DISTRIBUTION(k, t)` — распределение Пуассона, представляемое в виде  $e^{-t} \sum_{m=0}^k \frac{t^m}{m!}$ ;
- `BINOMIAL_DENSITY(k, n, p)` — плотность вероятности биномиального распределения, в виде  $\frac{P^k (1-p)^{n-k}}{k!(n-k)!} n!$ ;
- `BINOMIAL_DISTRIBUTION(k, n, p)` — биномиальное распределение, представляемое в виде суммы ряда;
- `HYPERGEOMETRIC_DENSITY(k, n, p)` — плотность вероятности гипергеометрического распределения;
- `HYPERGEOMETRIC_DISTRIBUTION(k, n, p)` — гипергеометрическое распределение;
- `STUDENT(t, v)` — плотность вероятности распределения Стьюдента;
- `BETA(x, y)` — бета-функция (Эйлеров интеграл первого рода) представляется в виде следующего интеграла:

$$B(x, y) = 2 \int_0^1 t^{2x-1} (1-t^2)^{y-1} dt.$$

Бета-функция имеет большое количество функциональных представлений. При расчетах наиболее часто она представляется через гамма-функции:

$$B(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}.$$

Рассмотрим примеры вычисления вероятностных функций.

*Пример 3.5.* Пусть необходимо получить аналитические и численные значения рассмотренных выше вероятностных функций.

Аналитические и численные значения вероятностных функций на экране имеют вид, показанный на рис. 3.10.

```

#1:  LOAD(C:\DW5Trial\DW\MATH\PROBABIL.MTH)
#2:  Ψ(x)

#3:  
$$\text{LN}(x) = \frac{2 \cdot x \cdot \int_0^{\infty} \frac{t_-}{(t_-^2 + x^2) \cdot (\hat{e}^{\pi \cdot t_-} - 1)} dt_- - 2 \cdot x \cdot \int_0^{\infty} \frac{t_-}{(t_-^2 + x^2) \cdot (\hat{e}^{\pi \cdot t_-} + 1)} dt_-}{2 \cdot x} + 1$$


#4:  Ψ(2.5)
#5:  0.7031566406
#6:  POISSON_DENSITY(k, t)

#7:  
$$\frac{\hat{e}^{-t} \cdot t^k}{k!}$$

#8:  POISSON_DENSITY(3, 1)
#9:  0.06131324019
#10: POISSON_DISTRIBUTION(k, t)

#11: 
$$\hat{e}^{-t} \cdot \sum_{n=0}^{\infty} \frac{t^n}{n!}$$

#12: POISSON_DISTRIBUTION(3, 1)
#13: 0.9810118431
#14: BINOMIAL_DENSITY(k, n, p)

#15: 
$$\frac{p^k \cdot (1-p)^{n-k} \cdot n!}{k! \cdot (n-k)!}$$

#16: BINOMIAL_DENSITY(2, 7, 0.5)
#17: 0.1640625
#18: BINOMIAL_DISTRIBUTION(r, n, p)

#19: 
$$(1-p)^n \cdot n! \cdot \sum_{m=0}^{\text{FLOOR}(-\text{ABS}(n-r)/2 + n/2 + r/2)} \frac{p^m \cdot (1-p)^{-m}}{m! \cdot (n-m)!}$$

#20: BINOMIAL_DISTRIBUTION(2, 7, 0.5)
#21: 0.2265625
#22: HYPERGEOMETRIC_DENSITY(k, n, p)

#23: 
$$\frac{n! \cdot p! \cdot (j-n)! \cdot (j-p)!}{j! \cdot k! \cdot (j+k-n-p)! \cdot (n-k)! \cdot (p-k)!}$$

#24: HYPERGEOMETRIC_DENSITY(1, 2, 3)

#25: 
$$\frac{6 \cdot (j-3)}{j \cdot (j-1)}$$

#26: STUDENT(t, v)

```

**Рис. 3.10.** Аналитические и численные значения вероятностных функций (*начало*)

$$\begin{aligned} \#27: & - \frac{\left(\frac{v}{2} + v\right)^{v/2} \cdot \left(\frac{v}{2} - \frac{1}{2}\right)!}{\sqrt{\pi} \cdot \left(\frac{v}{2}\right)!} + \\ & \frac{\left(\frac{v}{2} - \frac{1}{2}\right)! \cdot \int_0^{v/(t^2 + v)} \frac{t_{-}^{(v-2)/2} \cdot (\sqrt{(1-t_{-})} - 1)}{\sqrt{(1-t_{-})}} dt_{-}}{\sqrt{\pi} \cdot \left(\frac{v}{2} - 1\right)!} + 1 \\ \#28: & \text{STUDENT}(0.5, 2) \\ \#29: & 0.3333333333 \\ \#30: & \beta(x, y) \\ \#31: & \frac{(x-1)! \cdot (y-1)!}{(x+y-1)!} \\ \#32: & \beta(0.5, 1) \\ \#33: & 2 \end{aligned}$$

Рис. 3.10 (окончание)

В файле PROBABIL.MTN, кроме статистических, находится ряд других функций, представляющих для пользователей интерес в смысле их получения, в среде Derive 5, в виде векторов. Такими являются следующие две функции:

□ FIBONACCI(n) — функция Фибоначчи (числа Фибоначчи), вычисляемые по выражению:

$$\left( \begin{vmatrix} 0 & 1 \\ 1 & 1 \end{vmatrix}^n \cdot [0,1] \right)_1;$$

□ CATALAN(n) — функция Каталана, определяемая в Derive 5 как:

$$\frac{2^{2n} \left(n - \frac{1}{2}\right)!}{\sqrt{n} (n+1)!}.$$

Ниже приведены результаты табулирования функции Фибоначчи и Каталана с помощью функции VECTOR (рис. 3.11). В строках #1,

#2, #5, #6 получены математические выражения функций, в остальных строках — векторы чисел Фибоначчи и Каталана:

#1: FIBONACCI(n)

$$\#2: \left( \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}^n \cdot \begin{bmatrix} 0 & 1 \end{bmatrix} \right)_1$$

#3: VECTOR(FIBONACCI(n), n, 1, 10)

#4: [1, 1, 2, 3, 5, 8, 13, 21, 34, 55]

#5: CATALAN(n)

$$\#6: \frac{2^{2 \cdot n} \cdot \left( n - \frac{1}{2} \right)!}{\sqrt{\pi} \cdot (n+1)!}$$

#7: VECTOR(CATALAN(n), n, 1, 10)

#8: [1, 2, 5, 14, 42, 132, 429, 1430, 4862, 1.6796 · 10<sup>4</sup>]

**Рис. 3.11.** Табулирование функций Фибоначчи и Каталана с помощью функции VECTOR

## Интегральные показательные функции

Рассмотрим основные интегральные показательные функции системы Derive 5.

□  $Ei(x, m)$  — интегральная показательная функция

$$Ei(x, m) = - \int_{-x}^{\infty} \frac{e^{-t}}{t} dt, \text{ вычисляемая методом разложения в ряд}$$

при числе членов разложения, равном  $m$ .

□  $Li(x, m)$  — интегральный логарифм  $li = \int_0^x \frac{dt}{\ln t} = Ei(\ln(x)), x > 0$ .

□  $Si(x)$  — интегральный синус  $\sin x = - \int_x^{\infty} \frac{\sin t}{t} dt = - \frac{\pi}{2} + \int_0^x \frac{\sin t}{t} dt$ .

□  $Ci(x)$  — интегральный косинус  $Ci(x) = - \int_x^{\infty} \frac{\cos t}{t} dt = \int_0^x \frac{\cos t}{t} dt$ .

□  $En(n, x)$  — интегральная показательная функция порядка  $n$ ,

определяемая в виде  $En = \int_1^{\infty} \frac{e^{-xt}}{t^n} dt, \operatorname{Re}(x) > 0, n \geq 0$ .

#1:	euler_gamma	
#2:		0.5772156649
#3:	EI(x, m)	
#4:		$\text{LN}(x) + \text{euler\_gamma} + \sum_{n=1}^m \frac{x^{n-1}}{n-1 \cdot n!}$
#5:	EI(1, 2)	
#6:		1.827215664
#7:	LI(x, m)	
#8:		$\text{LN}(\text{LN}(x)) + \text{euler\_gamma} + \sum_{n=1}^m \frac{\text{LN}(x)^{n-1}}{n-1 \cdot n!}$
#9:	LI(2, 1)	
#10:		0.9038499248
#11:	SI(x)	
#12:		$\int_0^x \frac{\text{SIN}(t_-)}{t_-} dt_-$
#13:	SI(2.5)	
#14:		1.778520173
#15:	CI(x)	
#16:		$\text{LN}(x) + \text{euler\_gamma} + \int_0^x \left( \frac{\text{COS}(t_-)}{t_-} - \frac{1}{t_-} \right) dt_-$
#17:	CI(2.5)	
#18:		0.2858711963
#19:	EN(n, x)	
#20:		$\int_1^{\infty} \frac{t_-^x}{t_-^n} dt_-$
#21:	EN(3, 1.5)	
#22:		0.05673949017

Рис. 3.12. Расчет значений интегральных показательных функций

На рис. 3.12 показаны расчетные формулы вычисления функций и их численные значения для конкретных данных. В формулы функций входит константа Эйлера. Ее значение вычисляется по команде `euler_gamma` и находится во второй строке. В правильности вычисления функций можно убедиться путем непосредственных вычислений методом подстановки в формулы исходных данных (команда **Sub** панели инструментов).

## Интегралы Френеля

Интегралы Френеля имеют вид:

$$\Phi(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{t^2} dt;$$

$$S(x) = \frac{2}{\sqrt{2\pi}} \int_0^x \sin t^2 dt;$$

$$C(x) = \frac{2}{\sqrt{2\pi}} \int_0^x \cos t^2 dt.$$

Функции вычисления интегралов Френеля в системе Derive 5 имеют вид:

- `FRESNEL_SIN(x)` — синусный интеграл Френеля;
- `FRESNEL_SIN_SERIES(x, n)` — синусный интеграл Френеля, представленный в виде ряда с числом членов, равным  $n$ ;
- `FRESNEL_COS(x)` — косинусный интеграл Френеля;
- `FRESNEL_COS_SERIES(x, n)` — косинусный интеграл Френеля, представленный в виде ряда с числом членов, равным  $n$ .

На рис. 3.13 показаны формулы определения функций и их расчетные численные значения при  $x = 1,27$  и  $n = 5$ .

#1: FRESNEL\_SIN(x)

$$\int_0^x \sin\left(\frac{\pi \cdot t_-^2}{2}\right) dt_-$$

#3: FRESNEL\_SIN(1.27)

#4: 0.6707201392

#5: FRESNEL\_SIN\_SERIES(x, n)

$$\frac{\pi \cdot x^3 \cdot \sum_{k_-=0}^n \frac{x^{4 \cdot k_-} \cdot \left(\frac{\pi}{2}\right)^{2 \cdot k_-} \cdot \cos(\pi \cdot k_-)}{(4 \cdot k_- + 3) \cdot (2 \cdot k_- + 1)!}}{2}$$

#7: FRESNEL\_SIN\_SERIES(1.27, 5)

#8: 0.6707188357

#9: FRESNEL\_COS(x)

$$\int_0^x \cos\left(\frac{\pi \cdot t_-^2}{2}\right) dt_-$$

#11: FRESNEL\_COS(1.27)

#12: 0.6641459655

#13: FRESNEL\_COS\_SERIES(x, n)

$$x \cdot \sum_{k_-=0}^n \frac{x^{4 \cdot k_-} \cdot \left(\frac{\pi}{2}\right)^{2 \cdot k_-} \cdot \cos(\pi \cdot k_-)}{(4 \cdot k_- + 1) \cdot (2 \cdot k_-)!}$$

#15: FRESNEL\_COS\_SERIES(1.27, 5)

#16: 0.6641387679

Рис. 3.13. Расчет интегралов Френеля

## Функции Бесселя

Функции Бесселя первого и второго рода имеют вид:

□ BESSEL\_J(n, x) — функция Бесселя первого рода ( $n$  — рациональное число);

□ BESSEL\_Y(n, x) — функция Бесселя второго рода (при  $n$  целом).

```

#1:  BESSEL_J(1, 2.7)
#2:                                     0.4416013791
#3:  BESSEL_Y(1, 2.7)
#4:                                     0.2276324458

```

**Рис. 3.14.** Вычисление функций Бесселя

На рис. 3.14 приведены численные примеры вычисления функций Бесселя.

## Функции Эйри

Функции Эйри имеют вид:

- `AI_SERIES(x, n)` — функция Эйри  $Ai$ ;
- `BI_SERIES(x, n)` — функция Эйри  $Bi$ .

Обе функции вычисляются путем разложения в ряд с числом членов ряда, равном  $n$ .

На рис. 3.15 приведены результаты вычисления функции при  $x = 2,3$ ;  $n = 5$ .

```

#1:  AI_SERIES(2.3, 5)
#2:                                     0.02182462006
#3:  BI_SERIES(2.3, 5)
#4:                                     4.885036983

```

**Рис. 3.15.** Вычисление функций Эйри

## Ортогональные полиномы

Функции вычисления ортогональных полиномов Чебышева, Лежандра и Эрмита в системе Derive 5 имеют вид:

- `CHEBYCHEV_T(n, x)` — полином Чебышева первого рода;
- `CHEBYCHEV_U(n, x)` — полином Чебышева второго рода;
- `LEGENDRE_P(n, x)` — полином Лежандра;
- `HERMITE_H(n, x)` — полином Эрмита.



На рис. 3.16 приведены формулы полиномов для  $n = 1, 2, \dots, 6$ . Формулы получены путем табулирования функций ортогональных полиномов с помощью функции VECTOR.

```
#1:  CHEBYCHEV_I(n, x)
#2:  VECTOR(CHEBYCHEV_I(n, x), n, 1, 6, 1)
#3:  [x, 2·x2 - 1, 4·x3 - 3·x, 8·x4 - 8·x2 + 1, 16·x5 - 20·x3 + 5·x, 32·x6 - 48·x4 + 18·x2 - 1]
#4:  CHEBYCHEV_U(n, x)
#5:  VECTOR(CHEBYCHEV_U(n, x), n, 1, 6, 1)
#6:  [2·x, 4·x2 - 1, 8·x3 - 4·x, 16·x4 - 12·x2 + 1, 32·x5 - 32·x3 + 6·x, 64·x6 - 80·x4 + 24·x2 - 1]
#7:  LEGENDRE_P(n, x)
#8:  VECTOR(LEGENDRE_P(n, x), n, 1, 6, 1)
#9:  [x, 0.5·(3·x2 - 1), 0.5·x·(5·x2 - 3), 0.125·(35·x4 - 30·x2 + 3), 0.125·x·(63·x4 - 70·x2 + 15), 0.0625·(231·x6 - 315·x4 + 105·x2 - 5)]
#10: HERMITE_H(n, x)
#11: VECTOR(HERMITE_H(n, x), n, 1, 6, 1)
#12: [2·x, 4·x2 - 2, 8·x3 - 12·x, 16·x4 - 48·x2 + 12, 32·x5 - 160·x3 + 120·x, 64·x6 - 480·x4 + 720·x2 - 120]
```

**Рис. 3.16.** Вычисление ортогональных полиномов

Вычисление значений полиномов очевидно.

## Дзета-функция Римана, функция Гурвица

Эти функции имеют вид:

□  $ZETA(s, m)$  — дзета-функция Римана;

□  $NURWITZ\_ZETA(s, a, m)$  — функция Гурвица;

□  $LERCH\_PHI(x, s, s, m)$  — пси-функция.

На рис. 3.17 приведены формулы функций Гурвица и пси-функции. Дзета-функцию Derive 5 не выдает. Значения функций получены при следующих значениях аргументов:  $s = 0,4$ ;  $a = 2$ ;  $m = 5$ ;  $x = 0,5$ .

```

#1:  ζ(s, m)
#2:  ζ(0.5, 3.5)
#3:  -3.468064318
#4:  HURWITZ_ZETA(s, a, m)
#5:  - ζ(s, a + m + 1) + ζ(s, a) +  $\frac{2^{s-1} \cdot (2 \cdot a + 2 \cdot m + 1)^{1-s}}{s-1}$ 
#6:  HURWITZ_ZETA(0.4, 2, 5)
#7:  -2.133806927
#8:  LERCH_PHI(x, s, a, m)
#9:   $\sum_{k=0}^m x^k \cdot (k+a)^{-s}$ 
#10: LERCH_PHI(0.5, 0.4, 2, 5)
#11: 1.334176881

```

**Рис. 3.17.** Расчет функций Гурвица и пси-функции

## Функция инверсная

Функция имеет вид:  $\text{INVERSE}(y, x)$  — инвертирование функции  $y = (x)$ , то есть образование функции  $x = (y)$ .

На рис. 3.18 показаны примеры реализации функции для случаев:

- ☐  $y(x) = 2^x - 4x$ ;
- ☐  $y(x) = 2 - 4x$ ;
- ☐  $y(x) = e^x$ ;
- ☐  $y(x) = \sin(x)$ ;
- ☐  $y(x) = x^2 - 2x + 1$ .

### Примечание

Если решения нет, то откликом будет знак вопроса (?).

#1:	$\text{INVERSE}(2^x - 4 \cdot x, x)$	
#2:		?
#3:	$\text{INVERSE}(2 - 4 \cdot x, x)$	
#4:		$\frac{2 - x}{4}$
#5:	$\text{INVERSE}(e^x, x)$	
#6:		$\text{LN}(x)$
#7:	$\text{INVERSE}(\text{SIN}(x), x)$	
#8:		$\text{ASIN}(x)$
#9:	$\text{INVERSE}(x^2 - 2 \cdot x + 1, x)$	
#10:		$\sqrt{x} + 1$

Рис. 3.18. Вычисление функции INVERSE

## Вычисление факториала

Факториал — это гамма-функция, имеющая вид —

$$\Gamma(n) = \int_0^{\infty} e^{-t} t^{n-1} dt \quad \text{при условии, что } n \text{ является числом целым.}$$

Гамма-функция имеет множество интегральных представлений. Справедливыми являются следующие соотношения:

$$\square \quad \Gamma(n+1) = n\Gamma(n);$$

$$\square \quad \Gamma(1) = \Gamma(2) = 1;$$

$$\square \quad \Gamma(n) = (n-1)!;$$

$$\square \quad \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi};$$

$$\square \quad \Gamma\left(-\frac{1}{2}\right) = 2\sqrt{\pi};$$

$$\square \quad \Gamma\left(n + \frac{1}{2}\right) = \sqrt{\pi / 2^n} (2n-1)!! = \left(n - \frac{1}{2}\right)!.$$

В системе Derive 5 гамма-функция и факториал отождествляются. Они представляются на экране в одной из следующих форм:  $n!$ , или  $\text{GAMMA}(n)$ , или  $\Gamma(n)$ . Гамма-функция, а значит и факториал, существует для случая целого и дробного числа, положитель-

ного и отрицательного, действительного и комплексного. Вычислительные процедуры имеют вид:

```
#1:  5!
#2:                                120
#3:   $\Gamma(6)$ 
#4:                                120
#5:  3.45!
#6:                                10.8547
#7:   $\Gamma(4.45)$ 
#8:                                10.8547
#9:  -3.45!
#10:                               -10.8547
#11:   $(3+4.5i)!$ 
#12:                               0.539971+0.131856 i
#13:   $\Gamma(4+4.5i)!$ 
#14:                               0.539971+0.131856 i
#15:   $\Gamma(3+4.5i)!$ 
#16:                               0.0756672-0.0695487i
```

При практических вычислениях не следует забывать, что  $\Gamma(n) = (n-1)!$ . Если необходимо вычислить факториал числа 5, то это можно сделать одним из следующих способов: ввести выражение 5!, вычислить GAMMA(6) (но не GAMMA(5)). Это видно из примера, при вычислении факториала комплексного числа  $3+4,5i$  двумя способами.

### 3.3. Вычисление функций при ограничивающих условиях

При решении практических задач иногда приходится вычислять одну из множества функций в зависимости от некоторых условий. Такие вычисления реализуются в системе Derive 5 с помощью функции IF, которая имеет следующие две формы:

□  $\text{IF}(r, y1, y2);$

□  $\text{IF}(r, y1, y2, y3).$

Функции выполняют следующие действия: если условие  $r$  истинно, то вычисляется функция  $y_1$ , в противном случае  $y_2$ ; если же истинность условия неизвестна, то вычисляется функция  $y_3$ .

*Пример 3.6.* Пусть необходимо вычислить функции:  $y_1 = x!$ ,  $y_2 = e^{2x} + 3x^2 - 2$ , при этом по условию задачи первая из них вычисляется при  $x \geq 1$ , вторая — в противном случае. При определении порядка решения функция IF имеет вид —  $IF(x \geq 1, y_1 = x!, y_2 = e^{2x} + 3x^2 - 2)$ . Если теперь  $x=3$ , то ответом будет  $y_1 = 3! = 6$ , а если  $x=0$ , то  $y_2 = e^{0} + 3 \cdot 0^2 - 2 = -1$ .

*Пример 3.7.* Теперь изменим условие задачи. Пусть анализируемые функции имеют вид:  $y_1 = 2x^2 + 1$ ,  $y_2 = (x+1)/(x-1)$ ,  $y_3 = \ln(2x)$ . Необходимо вычислить значения  $y$  при  $x=0,5$ ;  $x=8$ ;  $x=a$ . Результатами вычислений с помощью функции  $IF(x \leq 1, y_1, y_2, y_3)$  будут:

□ при  $x = 0,5$  —  $y_1 = 2x^2 + 1 = 1,5$ ;

□ при  $x = 8$  —  $y_2 = (x+1)/(x-1) = 1,3333\dots$ ;

□ при  $x = a$  —  $y_3 = \ln(2a) = \ln(2) + \ln(a) = \ln(a) + 0,693147$ .

Из описания функции IF видно, что она аналогична операторам ветвления в универсальных языках программирования.

Рациональная технология использования оператора IF состоит в выполнении следующих действий:

- ввод вычисляемых выражений (в нашем случае на экране появятся следующие выражения: #1:  $2x^2 + 1$ , #2:  $(x+1)/(x-1)$ , #3:  $\ln(2x)$ );
- ввод функции IF в следующем виде:  $IF(x \leq 1, \#1, \#2, \#3)$  (на экране будет функция IF, в которой, вместо номеров строк, сами выражения);
- ввод значений аргументов  $x$  с помощью команды **Sub** на панели инструментов.

При нажатии кнопки **Simplify** получаем решение, которое на экране имеет вид, показанный на рис. 3.19.

$$\begin{array}{ll}
\#1: & 2 \cdot x^2 + 1 \\
\#2: & \frac{x + 1}{x - 1} \\
\#3: & \text{LN}(2 \cdot x) \\
\#4: & \text{IF} \left( x \leq 1, 2 \cdot x^2 + 1, \frac{x + 1}{x - 1}, \text{LN}(2 \cdot x) \right) \\
\#5: & \frac{3}{2} \\
\#6: & \frac{9}{7} \\
\#7: & \text{LN}(2 \cdot a) \\
\#8: & \text{LN}(a) + 0.6931471805
\end{array}$$

**Рис. 3.19.** Пример использования функции IF

Функцию IF целесообразно использовать, если требуются вычисления функции при большом числе значений аргументов. Рассмотрим типичный для этого случая пример.

*Пример 3.8.* Пусть необходимо табулировать функцию  $y = e^x + 1$ , в диапазоне от 0 до 2, с шагом  $h_1$  или, в диапазоне от  $-0,1$  до  $-1$ , с шагом  $h_2$ . При этом шаг может меняться. Функции табуляции для этих случаев будут иметь вид:

$$\square \text{ VECTOR}([x, e^x + 1], x, 0, 2, h);$$

$$\square \text{ VECTOR}([x, e^x + 1], x, -0.1, -1, h).$$

Теперь используем IF для выбора функции табуляции в зависимости от шага  $h$ :

$$\text{IF}(h \geq .2, \text{VECTOR}([x, e^x + 1], x, 0, 2, h), \text{VECTOR}([x, e^x + 1], x, -0.1, -1, h)).$$

Задавая значения  $h$ , получим результаты табулирования по одной из функций табуляции. Решение для случая  $h=0.25$  и  $h=-0.1$  показано на рис. 3.20.

```

#1:   $\hat{e}^x + 1$ 
#2:  VECTOR( $[x, \hat{e}^x + 1]$ , x, 0, 2, h)
#3:  VECTOR( $[x, \hat{e}^x + 1]$ , x, -0.1, -1, h)
#4:  IF( $h \geq 0.2$ , VECTOR( $[x, \hat{e}^x + 1]$ , x, 0, 2, h), VECTOR( $[x, \hat{e}^x + 1]$ , x, -0.1, -1, h))

```

#5:

0	2
0.25	2.284025416
0.5	2.648721270
0.75	3.117000016
1	3.718281828
1.25	4.490342957
1.5	5.481689070
1.75	6.754602675
2	8.389056098

#6:

-0.1	1.904837418
-0.2	1.818730753
-0.3	1.740818220
-0.4	1.670320046
-0.5	1.606530659
-0.6	1.548811636
-0.7	1.496585303
-0.8	1.449328964
-0.9	1.406569659
-1	1.367879441

**Рис. 3.20.** Табулирование функции при шаге 0,25 и -0,1

## 3.4. Системы счисления

В системе Derive 5 можно выполнять математические операции в различных системах счисления, в том числе в двоичной, восьмеричной и шестнадцатеричной, которые широко используются в вычислительной технике.

Система позволяет:

- ☐ представлять числа в различных системах счисления (СС);
- ☐ переводить числа из одной системы счисления (СС) в другую;
- ☐ выполнять арифметические операции в различных СС;
- ☐ определять погрешности вычислений, связанные с неточностью перевода чисел из одной СС в другую.

Все это дает возможность активно изучать арифметические основы ЭВМ на упражнениях и при выполнении лабораторных работ по предмету "Информатика".

Рассмотрим технологии вычислений на примерах.

### 3.4.1. Представление чисел в системах счисления

Для представления числа, заданного в системе счисления с основанием  $n$ , необходимо настроить систему Derive 5 на ввод и вывод чисел в одну из следующих систем счисления: десятичную (**Decimal**), двоичную (**Binary**), восьмеричную (**Octal**), шестнадцатеричную (**Hexadecimal**).

Технология настройки состоит в следующем:

- ☐ щелкнуть мышью по пункту **Declare | Input** главного меню (на экране появляется окно **Input Settings**, показанное на рис. 3.21);
- ☐ на панели **Input Mode** устанавливается режим численных вычислений (**Character**), в строке ввода **Radix** указывается необходимая система счисления;
- ☐ после нажатия кнопки **OK** на экране появляется сообщение о системе счисления, используемой при вводе чисел;
- ☐ при установлении типа СС, для выводимых чисел, повторяются первые два пункта, при этом на первом шаге активизируется пункт **Declare | Output** главного меню (на экране появляется окно **Output Settings**, показанное на рис. 3.22, в котором устанавливается тип СС выводимых чисел;
- ☐ после нажатия кнопки **OK** на экране появляется сообщение о СС выводимых чисел.



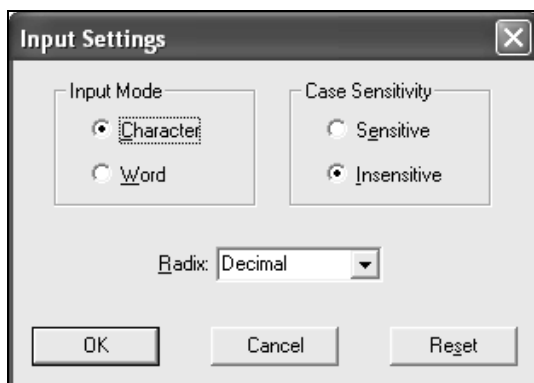


Рис. 3.21. Окно настройки системы счисления при вводе чисел

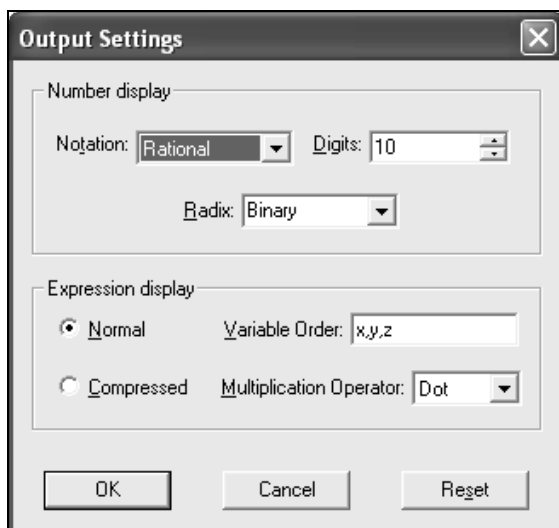


Рис. 3.22. Окно настройки системы счисления при выводе чисел

*Пример 3.9.* Пусть необходимо перевести число  $N_{10} = 123,175$ , заданное в десятичной СС, в двоичную  $N_2$ , восьмеричную  $N_8$  и шестнадцатеричную  $N_{16}$ . Предполагается, что система настроена на десятичную СС.

Технология решения задачи в системе Derive 5 следующая:

- ❑ набор и ввод числа 123.175 (на экране и в строке пользователя отображается число 123.175);
- ❑ настройка системы на перевод числа в двоичную СС (на экране выводится сообщение — **Output Base := Binary**);
- ❑ ввод значения (<Enter>) — на экране отображается двоичное число 1111011.001;
- ❑ настройка системы на перевод числа в восьмеричную СС (на экране выводится сообщение — **Output Base := Octal**);
- ❑ ввод значения (<Enter>) — на экране отображается восьмеричное число 173.1314631;
- ❑ настройка системы на перевод числа в шестнадцатеричную СС (на экране выводится сообщение — **Output Base := Hexadecimal**);
- ❑ ввод значения (<Enter>) — на экране отображается число в шестнадцатеричное число 7B.2CCCCCCC.

Процедуры преобразования числа из десятичной СС в двоичную, восьмеричную и шестнадцатеричную показаны на рис. 3.23.

```

#1:  OutputBase := Decimal
#2:  123.175
#3:  OutputBase := Binary
#4:  1111011.001
#5:  OutputBase := Octal
#6:  173.1314631
#7:  OutputBase := Hexadecimal
#8:  7B.2CCCCCCC

```

**Рис. 3.23.** Перевод чисел в двоичную, восьмеричную и шестнадцатеричную системы счисления

### 3.4.2. Перевод чисел из одной СС в другую

Для перевода числа  $N_n$ , представленного в системе счисления с основанием  $n$ , в число  $N_m$ , представленного в системе

счисления с основанием  $m$ , необходимо настроить ввод на систему счисления с основанием  $n$ , а вывод — на систему счисления с основанием  $m$ . Далее требуется набрать исходное число и нажать клавишу <Enter>. В результате этих действий на экране появится число, представленное в новой системе счисления с основанием  $m$ .

*Пример 3.10.* Пусть требуется перевести число  $N_2 = 100111,011$ , представленное в двоичной СС, в десятичную  $N_{10}$ , восьмеричную  $N_8$  и шестнадцатеричную  $N_{16}$  системы счисления.

Компьютерная технология решения задачи заключается в выполнении следующих действий:

- ☐ настроить ввод и вывод на двоичную систему счисления;
- ☐ набрать двоичное число в строке ввода и нажать клавишу <Enter> (на экране число 100111.011);
- ☐ настроить вывод на десятичную СС (**Decimal**);
- ☐ нажать клавишу <Enter> (на экране появится ответ — 39.375);
- ☐ настроить вывод на восьмеричную СС (**Octal**);
- ☐ нажать клавишу <Enter> (на экране появится ответ — 47.3);
- ☐ настроить вывод на шестнадцатеричную СС (**Hexadecimal**);
- ☐ нажать клавишу <Enter> (на экране появится ответ — 27.6).

Вид отображаемой на экране информации показан на рис. 3.24.

```
#1:  InputBase := Binary
#2:  OutputBase := Binary
#3:  100111.011
#4:  OutputBase := Decimal
#5:  39.375
#6:  OutputBase := Octal
#7:  47.3
#8:  OutputBase := Hexadecimal
#9:  27.6
```

**Рис. 3.24.** Перевод чисел из одной системы счисления в другую

### 3.4.3. Арифметические операции в различных системах счисления

Технология выполнения арифметических операций в Derive 5 не зависит от системы счисления. Она одинакова для десятичной, двоичной, восьмеричной и шестнадцатеричной систем счисления. Для их выполнения достаточно настроить Derive 5 на ввод и вывод чисел в нужной СС и выполнить требуемые операции.

*Пример 3.11.* Необходимо найти сумму следующих чисел —  $N_1 = 26,247$  и  $N_2 = 34,618$ , заданных в десятичной СС. Вычисления выполнить в десятичной, двоичной, восьмеричной и шестнадцатеричной системах счисления.

Предположим, что система настроена на десятичную СС. Тогда, после выполнения сложения чисел в десятичной СС, достаточно последовательно перестраивать систему на вывод чисел в двоичной, восьмеричной и шестнадцатеричной СС, нажать клавишу <Enter> и выполнить команду **Approximate**. На экране будет сформирована сумма чисел в указанной СС. Процедуры решения задачи будут иметь вид:

```
#1: 26.247+34.618
#2:                                     60.865
#3: OutputBase:= Binary
#4: 11010.00111+ 100010
#5:                                     111100.1101
#6: OutputBase:= Octal
#7: 32.17635544+42.47432477
#8:                                     74.67270243
#9: OutputBase:=Hexadecimal
#10 1A.3F3B645A + 22.9E353F7C
#11                                     3C.DD70A307
```

### 3.4.4. Решение задач в различных системах счисления

Технология решения задач едина и не зависит от системы счисления. Необходимо только настроить систему на ввод и вывод чисел в соответствующей СС.

*Пример 3.12.* Пусть необходимо получить таблицу функции  $y = 3,27x + \ln(5,7x) - 2,63$  в диапазоне аргумента от единицы до пяти с шагом 0,5. Решение необходимо выполнить в двоичной СС.

Технология решения задачи состоит в выполнении следующих действий.

- Представление функции в двоичной СС. Для этого необходимо перевести все числа функции в двоичную СС, пользуясь технологией, описанной выше, и ввести функцию с новыми кодами чисел.

#### Примечание

Эту процедуру можно упростить. Достаточно набрать функцию в исходном виде, когда числа представлены в десятичной СС. Затем настроить систему на вывод в двоичной СС и нажать клавишу <Enter>.

- Набор и ввод функции табуляции — `VECTOR([x, #2], x, 1, 5, 0.5)` (здесь предполагается, что функция находится в строке #2).
- Щелкнуть мышью по кнопке **Approximate** на панели инструментов.

В результате на экране появится функция, представленная в виде таблицы. Вычислительные процедуры, для этого примера, имеют следующий вид:

```
#1: OutputBase: Binary
```

```
#2: 11.01000101x +LN(11.10110011 x)-10.10100001
```

```
#3: VECTOR([x, 11.01000101x +LN(11.10110011 x)-  
10.10100001], x, 1, 101, 0.1)
```

1	1.111100101
1.1	11.11111101
10	101.1110100
10.1	111.1100010
11	1001.100101

```

11.1      1011.011000
100       1101.001001
100.1     1110.111001
101       10000.10100

```

Чтобы выполнять арифметические операции, необходимо знать таблицы сложения и умножения в данной СС. Такие таблицы легко получить с помощью системы Derive 5.

*Пример 3.13.* Пусть требуется создать таблицу умножения в восьмеричной СС. Для ее создания, в восьмеричной системе счисления, необходимо выполнить следующие действия:

- ☐ создать таблицу умножения в десятичной СС, в диапазоне цифр алфавита восьмеричной СС (1,2,3,4,5,6,7);
- ☐ настроить с помощью пункта меню **Declare** вывод чисел в восьмеричной системе счисления;
- ☐ получить решение путем нажатия клавиши <Enter>.

Технология получения решения состоит в выполнении следующих процедур:

- ☐ щелкнуть мышью по кнопке **Autor Matrix** на панели инструментов (на экране появляется окно **Matrix Setup**);
- ☐ на панели **Matrix** установить размер матрицы (7×7) в соответствии с диапазоном цифр восьмеричной СС;
- ☐ щелкнуть мышью по кнопке **OK** (появляется новое окно **Autor 7×7 matrix**);
- ☐ произвести ввод чисел в пустые ячейки матрицы таблицы умножения:

1	2	3	4	5	6	7
2	4	6	8	10	12	14
3	6	9	12	15	18	21
4	8	12	16	20	24	28
5	10	15	20	25	30	35
6	12	18	24	30	36	42
7	14	21	28	35	42	49

- ☐ нажать кнопку **ОК** (на экране таблица умножения);
- ☐ ввести в строку ввода таблицу умножения путем нажатия клавиши <F3>;
- ☐ настроить систему на вывод данных в восьмеричной СС;
- ☐ нажать клавишу <Enter> (на экране сформируется таблица умножения в восьмеричной СС).

На рис. 3.25 показана искомая таблица умножения в восьмеричной СС.

1	2	3	4	5	6	7
2	4	6	10	12	14	16
3	6	11	14	17	22	25
4	10	14	20	24	30	34
5	12	17	24	31	36	43
6	14	22	30	36	44	52
7	16	25	34	43	52	61

**Рис. 3.25.** Таблица умножения в восьмеричной системе счисления

Аналогично можно создать таблицы сложения и умножения в любой из рассмотренных ранее СС.

*Пример 3.14.* Пользуясь таблицей умножения в восьмеричной СС, требуется найти произведение чисел 26 и 14. Решение этой задачи будет иметь следующий вид:

26

26

\* 14

130

+ 26

410

## Погрешности вычислений в различных СС

Перевод дробных чисел из десятичной СС в двоичную и другие осуществляется с погрешностью. Это приводит к ошибкам вычислений при использовании компьютерных технологий решения задач, особенно в тех случаях, когда приходится решать задачи с большим числом арифметических операций. Система Derive 5 позволяет путем сравнения вычислений в различных СС увидеть ошибки и даже оценить погрешности вычислений. Для этого достаточно сравнить результаты вычислений в различных системах числения.

## 3.5. Финансовые функции

Любые финансово-экономические расчеты можно выполнять с помощью универсальных математических методов, реализованных в Derive 5. Наличие специальных функций облегчает процесс решения задачи. Примером их являются финансовые функции, связанные с расчетами вкладов. Основанием для таких расчетов является следующее уравнение:

$$v(1+i)^n + p(1+it) + \frac{(1+i)^n - 1}{i} + f = 0. \quad (3.1)$$

В уравнении приняты обозначения:

- $v$  — сумма вклада на момент выполнения расчетов;
- $i$  — процент накопления вклада (процентная ставка);
- $n$  — число регулярных вложений;
- $f$  — сумма вклада в будущем;
- $t$  — время вложения или получения денег со вклада;
- $p$  — величина вложения или изъятия денежных средств (пай) (если денежные средства вносятся, то  $p$  — отрицательно, если изымаются, то  $p$  — положительно).



Относительно переменных  $v$ ,  $p$ ,  $t$ ,  $f$  уравнение (3.1) является линейным, позволяющим получить формулы для этих переменных в явном виде и выполнить расчеты путем непосредственных вычислений. Однако относительно переменных  $i$  и  $n$  оно является нелинейным, поэтому его решение возможно лишь численными методами.

Система Derive 5 имеет функции, позволяющие решать задачи, связанные со вкладами, не решая уравнения (3.1). Такими функциями являются:

- $PVAL(i, n, p, f, t)$  — определяет имеющуюся сумму вклада на данный момент (значение  $v$ );
- $FVAL(i, n, p, v, t)$  — определяет сумму вклада в будущем (значение  $f$ );
- $PMT(i, n, v, f, t)$  — периодические платежи (значение  $p$ );
- $NPER(i, p, v, f, t)$  — периодическая платежная ставка (значение числа платежных периодов  $n$ );
- $RATE(n, p, v, f, t)$  — периодическая процентная ставка (значение  $i$ ).

Использование финансовых функций имеет следующие особенности.

- Время  $t$  вложения или получения денег со вклада считается равным 0, если платежи оформляются в конце периода вложения, и 1, если платежи оформляются в начале срока вложения. Значение  $0 < t < 1$  будут в том случае, если платежи оформляются в промежутке начала и конца периода вложения (пропорционально всему периоду).
- Функции дают решение в виде формул, если переменные заданы в символьном виде.
- Функции можно представлять в упрощенном виде, сокращая число переменных, по экономическому смыслу задачи (программа самостоятельно находит значения недостающих переменных).

Ниже приведены примеры решения задач финансовых расчетов по вкладам.

*Пример 3.15.* Пусть вкладчиком является пенсионер, получающий ежемесячно 2130 рублей, которые ему перечисляют в банк. Какую сумму он накопит в течение пяти лет, если процентная ставка составляет — 7% в год?

В данном случае следует применить функцию FVAL при следующих исходных данных:  $i = 7\%/12$ ,  $n = 5 \times 12$ ,  $p = -2130$ ,  $t = 1$ ,  $v = 2130$ .

Из постановки задачи ясно, что платеж происходит вначале каждого периода, поэтому  $t = 1$ . Сумма вклада  $v$  в данный момент очевидна, поэтому пай (вложение) и сумма вклада отличаются только знаками. Система Derive 5 "догадывается", что  $v = |p|$  и  $t = 1$ . Поэтому функцию FVAL можно представить в виде:

$$\text{FVAL}(i, n, p);$$

С учетом исходных данных:

$$\text{FVAL}(7\%/12, 5*12, -2130).$$

Решение задачи показано на рис. 3.26.

#1: FVAL(i, n, p, v, t)

$$\#2: \frac{p \cdot (i \cdot t + 1)}{i} - \frac{(i + 1)^n \cdot (i \cdot (p \cdot t + v) + p)}{i}$$

#3: FVAL(i, n, p)

$$\#4: \frac{p}{i} - \frac{p \cdot (i + 1)^n}{i}$$

$$\#5: \text{FVAL}\left(\frac{7\%}{12}, 60, -2130, 2130, 1\right)$$

$$\#6: 1.503628805 \cdot 10^5$$

$$\#7: \text{FVAL}\left(\frac{7\%}{12}, 60, -2130\right)$$

$$\#8: 1.524928805 \cdot 10^5$$

**Рис. 3.26.** Процедура решения задачи определения накопления

В строках #2 и #4 приведены формулы, являющиеся аналитическими решениями уравнения (3.1) относительно  $f$  в случае точного и упрощенного решения. В строках #5 и #7 находятся финансовые функции с числовыми данными примера, а в строках #6 и #8 — ответы. Из ответов следует, что пенсионер в течение пяти лет накопит 150 362 рубля 88 копеек. Этот ответ не совпадает с ответом, полученным по приближенной формуле на величину одного вклада (месячную пенсию пенсионера).

*Пример 3.16.* Пусть требуется определить, какой первоначальный вклад необходимо внести в банк, чтобы через 10 лет накопленная сумма была равна 100 000 рублей, если процентная ставка равна 9% годовых, а ежегодный вклад равен 1000 рублей.

В данном случае задача решается с помощью функции  $PVAL(i, n, p, f, t)$ , где  $i = 9\%$ ,  $n = 10$ ,  $p = 1000$ ,  $t = 1$ ,  $f = 10\,000$ . Решение задачи показано на рис. 3.27.

#1:  $PVAL(i, n, p, f, t)$

$$\#2: \frac{(i + 1)^{-n} \cdot (p \cdot (i \cdot t + 1) - f \cdot i)}{i} - \frac{p \cdot (i \cdot t + 1)}{i}$$

#3:  $PVAL(9\%, 10, -1000, -100000, 1)$

#4:  $4.923632758 \cdot 10^4$

**Рис. 3.27.** Процедура решения задачи определения стартового вклада для накопления установленной денежной суммы

Таким образом, чтобы накопленная сумма через 10 лет была равна 100 000 рублей, необходимо иметь первоначальный вклад 49 236 рублей и ежегодный 1000 рублей.

Система Derive 5 имеет несколько сотен различных функций. Такими, например, являются графические, демонстрационные и специальные функции, которые в этой главе не рассматривались, в связи с ограниченным объемом книги и редкими случаями их использования на практике (особенно в учебном процессе). С такими функциями можно ознакомиться в справочной системе Help. При этом трудностей в их применении не будет, так как технология расчетов практически не отличается от рассмотренной выше, при вычислении специальных функций.



# Специальные вычисления и преобразования математических функций

В этой главе рассматриваются наиболее часто встречающиеся на практике преобразования и вычисления математических функций: вычисление пределов, дифференцирование, разложение в ряд, интегрирование, суммирование, вычисление производных, табулирование. Эти вычисления в системе Derive 5 осуществляются с помощью функций, сосредоточенных в пункте меню **Calculus**. Такими функциями являются:

- ☐ **Limit** — определение пределов;
- ☐ **Differentiate** — вычисление производных;
- ☐ **Taylor Series** — разложение функции в ряд Тейлора;
- ☐ **Integrate** — определение интеграла функции;
- ☐ **Sum** — вычисление суммы членов ряда;
- ☐ **Product** — вычисление произведений членов ряда;
- ☐ **Vector** — табулирование функции и представление ее в виде вектора;
- ☐ **Table** — табулирование функции и представление ее в виде матрицы.

Технология вычисления интегралов, в связи с особой важностью этой функции, рассматривается в отдельной главе.

## 4.1. Табулирование функции

Табулирование функции  $f(x)$  может быть осуществлено с помощью функций **VECTOR** или **TABLE**.

Технология табулирования с помощью функции **VECTOR** состоит в следующем.

- Вводится табулируемая функция.
- Щелчок мышью по пункту меню **Calculus | Vector**. На экране появится окно **Calculus Vector** (рис. 4.1) с четырьмя полями ввода, в которые вводятся: имя аргумента функции (**Variable**); начальное  $x_n$  (**Starting Value**); конечное  $x_k$  (**Ending Value**) значения аргумента и шаг таблицы  $h$  (**Step Size**).

### Примечание

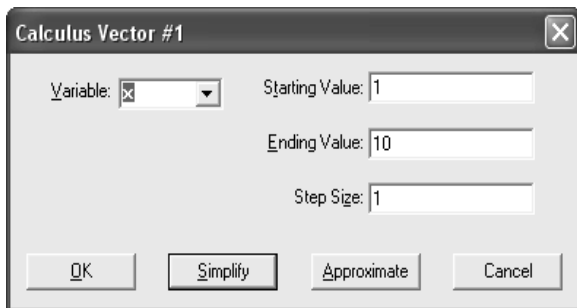
По умолчанию  $x_n = 1$ ,  $x_k = 10$ ,  $h = 1$ .

- Щелчок мышью по одной из четырех кнопок:
  - **OK** — вывод на экран функции табуляции;
  - **Simplify** — вывод на экран вектора табулируемой функции с точными значениями элементов вектора;
  - **Approximate** — вывод на экран вектора табулируемой функции с численными значениями элементов вектора;
  - **Cancel** — отказ от решения.

При решении конкретной задачи полезно воспользоваться кнопкой **OK** с выводом на экран функции табуляции. Это дает возможность проверить правильность выражения функции и, если необходимо, выполнить редактирование путем вызова ее в окно ввода (клавиша <F3>). Для получения решения после просмотра функции необходимо ее выделить и щелкнуть по кнопке **Simplify** или **Approximate** на панели инструментов.

Технология табулирования функции с помощью **Table** почти ничем не отличается от табулирования с помощью функции **Vector**. Отличие состоит лишь в конечном результате. Ответ будет в виде

матрицы с двумя столбцами, в которых указаны значение аргумента и значение функции.



**Рис. 4.1.** Окно ввода данных табуляции

На рис. 4.2 и 4.3 показаны результаты табулирования функции  $xe^x$  при  $x_n = 0$ ,  $x_k = 5$ ,  $h = 0,5$  с помощью функций VECTOR и TABLE.

$$x \cdot e^x$$

VECTOR( $x \cdot e^x$ ,  $x$ , 0, 5, 0.5)

[0, 0.8243606353, 2.718281828, 6.722533605, 14.77811219, 30.45623490, 60.25661076, 115.9040818, 218.3926001, 405.0770908, 742.0657955]

TABLE( $x \cdot e^x$ ,  $x$ , 0, 5, 0.5)

**Рис. 4.2.** Табулирование  $xe^x$  с помощью функции VECTOR

0	0
0.5	0.8243606353
1	2.718281828
1.5	6.722533605
2	14.77811219
2.5	30.45623490
3	60.25661076
3.5	115.9040818
4	218.3926001
4.5	405.0770908
5	742.0657955

**Рис. 4.3.** Результаты табулирования  $xe^x$  с помощью функции TABLE

Функции `VECTOR` и `TABLE` позволяют табулировать функции с переменным шагом, одновременно несколько функций с одним и тем же аргументом, а также функции с многими переменными.

При табулировании функции с переменным шагом необходимо в строке ввода **Starting Value** (окна **Calculus Vector**) записать вектор значений аргумента, например  $[1, 2.6, 6, 7.5, 12, \dots]$ , а в строках ввода **Ending Value** и **Step Size** записать 1. Дальнейшие процедуры имеют прежний вид.

При табулировании одновременно нескольких функций с одним и тем же аргументом необходимо вместо одной функции ввести вектор функций. Например, если нужно одновременно табулировать функции  $e^x$ ,  $\sin(x)$ ,  $\ln(x)$ , то вектор должен быть представлен в виде  $[e^x, \sin(x), \ln(x)]$ . Дальнейшие процедуры прежние.

Ниже показаны результаты табулирования этих функций при  $x_n = 0$ ,  $x_k = 2$ ,  $h = 0,25$  с помощью функций `VECTOR` и `TABLE` (рис. 4.4, 4.5).

`VECTOR([e^x, SIN(x), LN(x)], x, 0, 2, 0.25)`

1	0	LN(0)
1.284025416	0.2474039592	-1.386294361
1.648721270	0.4794255386	-0.6931471805
2.117000016	0.6816387600	-0.2876820724
2.718281828	0.8414709848	0
3.490342957	0.9489846193	0.2231435513
4.481689070	0.9974949866	0.4054651081
5.754602675	0.9839859468	0.5596157879
7.389056098	0.9092974268	0.6931471805

**Рис. 4.4.** Табулирование функций  $e^x$ ,  $\sin(x)$ ,  $\ln(x)$ , при  $x_n = 0$ ,  $x_k = 2$ ,  $h = 0,25$ , с помощью функции `VECTOR`

TABLE( $[e^x, \sin(x), \ln(x)], x, 0, 2, 0.25$ )

	0	1	0	LN(0)
0.25	1.284025416	0.2474039592	-1.386294361	
0.5	1.648721270	0.4794255386	-0.6931471805	
0.75	2.117000016	0.6816387600	-0.2876820724	
1	2.718281828	0.8414709848	0	
1.25	3.490342957	0.9489846193	0.2231435513	
1.5	4.481689070	0.9974949866	0.4054651081	
1.75	5.754602675	0.9839859468	0.5596157879	
2	7.389056098	0.9092974268	0.6931471805	

**Рис. 4.5.** Табулирование функций  $e^x$ ,  $\sin(x)$ ,  $\ln(x)$ ,  
при  $x_n = 0$ ,  $x_k = 2$ ,  $h = 0,25$ , с помощью функции TABLE

При табулировании функций можно не обращаться к пункту меню **Calculus | Vector** или **Calculus | Table**. Достаточно ввести эти функции с помощью клавиатуры компьютера в одном из следующих видов:

- ☐ VECTOR( $f(x), x, x_n, x_k, h$ ) — табулирование функции  $f(x)$  с постоянным шагом (ответ в виде вектора);
- ☐ VECTOR( $[x, f(x)], x, x_n, x_k, h$ ) — табулирование функции  $f(x)$  с постоянным шагом (ответ в виде матрицы);
- ☐ VECTOR( $[x, f(x)], x, [x_1, x_2, \dots, x_n]$ ) — табулирование функции  $f(x)$  с переменным шагом (ответ в виде матрицы);
- ☐ VECTOR( $[x, f_1(x), f_2(x), \dots, f_n(x)], x, x_n, x_k, h$ ) — табулирование множества функций с постоянным шагом, ответ в виде матрицы;
- ☐ VECTOR( $[x, f_1(x), f_2(x), \dots, f_n(x)], x, [x_1, x_2, \dots, x_n]$ ) — табулирование множества функций с переменным шагом (ответ в виде матрицы);
- ☐ TABLE( $f(x), x, x_n, x_k, h$ ) — табулирование функции  $f(x)$  с постоянным шагом (ответ в виде матрицы);
- ☐ TABLE( $f(x), x, [x_1, x_2, \dots, x_n]$ ) — табулирование функции  $f(x)$  с переменным шагом (ответ в виде матрицы);



- $\text{TABLE}(f_1(x), f_2(x), \dots, f_n(x), x, x_n, x_k, h)$  — табулирование множества функций с постоянным шагом (ответ в виде матрицы);
- $\text{TABLE}(f_1(x), f_2(x), \dots, f_n(x), x, [x_1, x_2, \dots, x_n])$  — табулирование множества функций с переменным шагом (ответ в виде матрицы).

После ввода функции табуляции необходимо щелкнуть мышью по одной из кнопок — **Simplify** или **Approximate** на панели инструментов.

**Vector** и **Table** позволяют табулировать функции многих аргументов. Для этого необходимо их применить многократно. Для случая функции двух переменных  $z = f(x, y)$  функция табуляции будет иметь вид:

$\text{Vector}([y, \text{Vector}([x, f(x, y)], x, x_n, x_k, h_x)], y, y_n, y_k, h_y);$

где:

- $f(x, y)$  — табулируемая функция двух аргументов;
- $x_n, x_k$  — начальное и конечное значения аргумента  $x$ ;
- $h_x$  — шаг аргумента  $x$ ;
- $y_n, y_k$  — начальное и конечное значения аргумента  $y$ ;
- $h_y$  — шаг аргумента  $y$ .

*Пример 4.1.* Пусть необходимо табулировать функцию  $z = f(x, y)$  при  $x_n = 0, x_k = 1, h_x = 0,1, y_n = 1, y_k = 5, h_y = 1$ .

Наиболее рациональной технологией решения задачи будет следующая.

- Ввод функции  $e^{xy}$  (на экране функция в строке #1).
- Ввод функции  $\text{VECTOR}([x, \#1], x, 0, 1, 0.1)$  (на экране, в строке #2, функция  $\text{VECTOR}([x, e^{xy}], x, 0, 1, 0.1)$ ).
- Ввод функции  $\text{VECTOR}([y, \#2], y, 1, 5, 1)$  (на экране функция —  $\text{VECTOR}([y, \text{VECTOR}([x, e^{xy}], x, 0, 1, 0.1)], y, 1, 5, 1)$ ).
- Щелчок мышью по кнопке **Approximate** на панели инструментов (на экране решение в виде пяти таблиц функции  $e^{xy}$ ).

На рис. 4.6 и 4.7 приведено решение только для двух значений  $y$  ( $y=1$  и  $y=2$ ).

$$e^{x \cdot y}$$

$$\text{VECTOR}([x, e^{x \cdot y}], x, 0, 1, 0.1)$$

$$\text{VECTOR}([y, \text{VECTOR}([x, e^{x \cdot y}], x, 0, 1, 0.1)], y, 1, 5, 1)$$

		0	1
		0.1	1.105170918
		0.2	1.221402758
		0.3	1.349858807
		0.4	1.491824697
1		0.5	1.648721270
		0.6	1.822118800
		0.7	2.013752707
		0.8	2.225540928
		0.9	2.459603111
		1	2.718281828

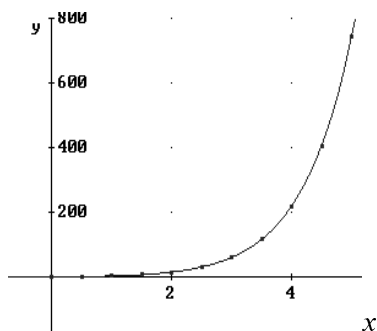
**Рис. 4.6.** Табуляция функции  $e^{xy}$  при  $y=1$

		0	1
		0.1	1.221402758
		0.2	1.491824697
		0.3	1.822118800
		0.4	2.225540928
2		0.5	2.718281828
		0.6	3.320116922
		0.7	4.055199966
		0.8	4.953032424
		0.9	6.049647464
		1	7.389056098

**Рис. 4.7.** Табуляция функции  $e^{xy}$  при  $y=2$

Технология табуляции с помощью функции `TABLE` аналогична рассмотренной выше для функции `VECTOR`.

Проверку правильности решения задачи можно выполнить методом визуализации результатов табуляции. Для этого необходимо с помощью функции **2D-plot window | Plot Expression** построить график табулируемой функции в диапазоне аргумента  $x$  (от  $x_n$  до  $x_k$ ). Затем построить график функции по данным таблицы табуляции. Если точки находятся на линии графика, то решение верно. На рис. 4.8 показано графическое представление результатов табулирования функции  $y = xe^x$ . Из рисунка видно, что решение задачи выполнено верно.



**Рис. 4.8.** Визуализация результатов табулирования функции  $y = xe^x$

## 4.2. Разложение функции в степенной ряд

### 4.2.1. Технология разложения функции в ряд Тейлора

Разложение функции  $y = f(x)$  в степенной ряд, в системе Derive 5, осуществляется по формуле Тейлора, которая имеет вид:

$$f(x) = f(a) + (x-a) \frac{f'(a)}{1!} + (x-a)^2 \frac{f''(a)}{2!} + \dots$$

$$\dots + (x-a)^n \frac{f^{(n)}(a)}{n!} + \dots \quad (4.1)$$

В формуле  $a$  — значение аргумента  $x$  функции  $y = f(x)$ , вокруг которого происходит разложение в ряд. Формула (4.1) справедлива для любого вещественного  $a$ , в том числе и для  $a = 0$ . При  $a = 0$  ряд Тейлора принимает вид:

$$f(x) = f(0) + x \frac{f'(0)}{1!} + x^2 \frac{f''(0)}{2!} + \dots + x^n \frac{f^{(n)}(0)}{n!} + \dots \quad (4.2)$$

Ряд (4.2) называется рядом *Маклорена*.

Разложение функций в степенной ряд реализуется в системе Derive с помощью функции `Taylor Series`. Эта функция имеет вид: `Taylor(f(x), x, x0, n)`;

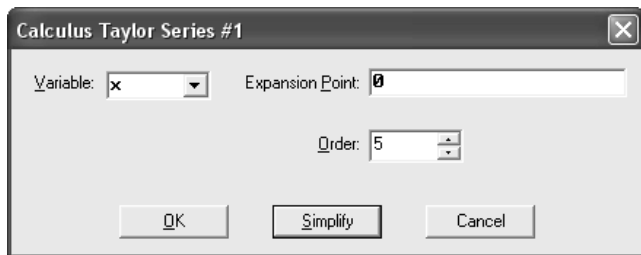
где:

- $f(x)$  — функция, разлагаемая в ряд;
- $x$  — аргумент функции  $y = f(x)$ ;
- $x_0$  — значение  $x$ , вокруг которого происходит разложение функции  $y = f(x)$ ,
- $n$  — число членов разложения.

Технология разложения функции в ряд Тейлора в системе Derive следующая:

- ввод функции  $y = f(x)$  (на экране функция  $y = f(x)$  в строке #1);
- выполнение команды **Calculus | Taylor Series** (на экране окно **Calculus Taylor Series** (рис. 4.9);
- в строке ввода **Variable** активизируется аргумент  $x$ , а в строках ввода **Expansion Point** и **Order** вводятся соответственно  $x_0$  и  $n$ ;
- выполнение команды **Simplify** (на экране формируется ответ — функция в виде степенного ряда).

На рис. 4.10 показано разложение в ряд функций  $y = \sin(x)$ ,  $y = \arcsin(x)$ ,  $y = \sinh(x)$  при  $x_0 = 0, n = 5$  по описанной выше технологии.



**Рис. 4.9.** Окно ввода данных при разложении функции в ряд

**SIN(x)**

**TAYLOR(SIN(x), x, 0, 5)**

$$\frac{x^5}{120} - \frac{x^3}{6} + x$$

**ASIN(x)**

**TAYLOR(ASIN(x), x, 0, 5)**

$$\frac{3 \cdot x^5}{40} + \frac{x^3}{6} + x$$

**SINH(x)**

**TAYLOR(SINH(x), x, 0, 5)**

$$\frac{x^5}{120} + \frac{x^3}{6} + x$$

**Рис. 4.10.** Разложение в ряд функций

$y = \sin(x), y = \arcsin(x), y = \sinh(x)$  при  $x_0 = 0, n = 5$

Обратим внимание на то, что степенные ряды состоят из трех членов, в то время как функция  $\text{TAYLOR}(f(x), x, 0, 5)$  имеет пять. Это объясняется тем, что функции являются нечетными и их четные члены при  $x = 0$  также равны нулю.

При практическом использовании формулы Тейлора у пользователя возникают три основных вопроса.

- ☐ Какие функции допускают разложение в ряд Тейлора?
- ☐ Каким числом членов ряда можно ограничиться в данном конкретном случае?
- ☐ Как найти погрешность разложения?

Ответим на эти вопросы, используя компьютерные технологии вычислений.

Разложить в ряд Тейлора можно любую функцию, имеющую  $n$  производных. Однако этим разложением не всегда можно воспользоваться. Это объясняется следующим свойством рядов:

- ряд может быть сходящимся или расходящимся;
- ряд Тейлора может не существовать при определенных значениях  $x_0$ . В качестве примера (рис. 4.11) приведены результаты разложения в ряд функций  $y = \ln(x)$ ,  $y = \sin(x)/x$  при  $x_0 = 0$ ,  $n = 7$  и  $x_0 = 1$ .

$$\begin{array}{l}
 \text{LN}(x) \\
 \text{TAYLOR}(\text{LN}(x), x, 0, 7) \\
 \quad ? \\
 \text{TAYLOR}(\text{LN}(x), x, 1, 7) \\
 \quad \frac{60 \cdot x^7 - 490 \cdot x^6 + 1764 \cdot x^5 - 3675 \cdot x^4 + 4900 \cdot x^3 - 4410 \cdot x^2 + 2940 \cdot x - 1089}{420} \\
 \frac{\text{SIN}(x)}{x} \\
 \text{TAYLOR}\left(\frac{\text{SIN}(x)}{x}, x, 0, 7\right) \\
 \quad -\frac{x^6}{5040} + \frac{x^4}{120} - \frac{x^2}{6} + 1
 \end{array}$$

**Рис. 4.11.** Разложение в ряд функций  $y = \ln(x)$ ,  $y = \sin(x)/x$  при  $x_0 = 0$ ,  $n = 7$  и  $x_0 = 1$

Из примера видно, что система Derive 5 не смогла разложить в ряд функцию  $\ln(x)$  при  $x_0 = 0$  и выполнила разложение при  $x_0 = 1$ . Причина здесь в том, что  $\ln(0)$  не существует. Функция  $\sin(x)/x$  при  $x = 0$  является неопределенностью вида  $\frac{0}{0}$ . Однако Derive 5 нашла решение. Это объясняется тем, что предел функции  $\sin(x)/x$  при  $x \rightarrow 0$  равен 1 и программа это определила. Вот это один из ярких примеров высокой интеллектуальности Derive 5.

Рассмотрим указанные выше свойства степенных рядов на примерах разложения в ряд Тейлора элементарных функций, при  $a = 0$ .

Разложение элементарных функций  $e^x, \sin(x), \cos(x)$  в ряд Маклорена приводит к степенным рядам следующего вида:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots; \quad (4.3)$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + (-1)^n \frac{x^{2n+1}}{(2n+1)!} + \dots; \quad (4.4)$$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!}. \quad (4.5)$$

При любом значении  $x$ , начиная с некоторого члена,  $n!$  растет быстрее, чем растет  $x^n$ , поэтому на основании признака Даламбера ряды (4.3), (4.4), (4.5) сходятся при любых значениях  $x$ . Функции  $\operatorname{tg}(x)$ ,  $\operatorname{ctg}(x)$  изменяются от  $-\infty$  до  $+\infty$ , поэтому степенные ряды этих функций являются расходящимися.

Все обратные тригонометрические функции удобно вычислять через  $\operatorname{arctg}(x)$ , для которого ряд Маклорена имеет вид:

$$\operatorname{arctg}(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots + (-1)^n \frac{x^{2n+1}}{2n+1} + \dots \quad (4.6)$$

Этот ряд сходится при  $|x| \leq 1$ . Его можно также использовать при

$x > 1$ , если воспользоваться соотношением:  $\operatorname{arctg}(x) = \frac{\pi}{2} - \operatorname{arctg} \frac{1}{x}$ .

Разложить в ряд Маклорена функцию  $y = \ln(x)$  нельзя, так как  $\ln(0) = -\infty$ , то есть степенной ряд является расходящимся. Для вычисления логарифма можно воспользоваться следующим выражением:

$$\ln(x+1) = x - \frac{x^2}{2} + \frac{x^3}{3} - \dots + (-1)^{n-1} \frac{x^n}{n} + \dots \quad (4.7)$$

Этот ряд сходится при  $-1 < x < 1$ , поэтому он позволяет вычислять натуральные логарифмы чисел от 0 до 2.

Вычислить логарифм чисел больше 2 можно при помощи следующего ряда:

$$\ln \frac{1+x}{1-x} = 2\left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots + \frac{x^{2n+1}}{2n+1} + \dots\right) \quad (4.8)$$

Этот ряд сходится также при  $x < 1$ , но при этом дробь  $\frac{1+x}{1-x}$  может принимать любое положительное значение.

Гиперболические функции выражаются через показательные. Они сходятся при любом значении  $x$ .

## 4.2.2. Погрешности степенных рядов Тейлора

В процессе вычисления функции бесконечный степенной ряд заменяется частичной суммой. В связи с этим следует знать, сколько членов ряда необходимо сохранить, чтобы вычислить функцию с заданной точностью. Решение этой задачи зависит от вида функции и значения аргумента.

### Случай 1. Ряд знакопеременный, члены ряда быстро убывают, $x < 1$

В этом случае ряд обладает следующим свойством: сумма отброшенных членов ряда, по абсолютной величине, не превышает значения последнего оставленного члена. Для данного случая определить число членов ряда довольно просто. Например, для ряда  $e^{-x}$  общий член ряда имеет вид —  $(-1)^n x^n / n!$ . Тогда число  $n$  можно найти из условия —  $x^n / n! \leq \xi$  (где  $\xi$  — погрешность вычисления функции). Если, например,  $\xi = 0,001$ , то  $x^n / n! \leq 0,0001$ . Тогда при  $x = 0,1$  достаточно оставить три члена ряда ( $n = 3$ ).

### Случай 2. Ряд знакопеременный, $x > 1$

В этом случае свойство частичной суммы ряда сохраняется, но ряд сходится медленно и необходимое число членов ряда должно



быть большим. Так, например, если в *случае 1* положить  $x = 1$ , то условием выбора числа членов ряда будет:  $1^n / n! \leq 0,001$  или  $n! = 1000$ , то есть  $n \approx 7$ .

### Случай 3. Ряд не знакопеременный

В этом случае общих правил выбора числа членов из условий точности не существует.

## 4.2.3. Компьютерные технологии оценки погрешностей рядов

### Способ 1. Табулирование функций

Сущность данного способа состоит в следующем: осуществляется табулирование исходной функции и полученной в результате разложения в ряд Тейлора. Начальное и конечное значения  $x$  выбираются из диапазона разложения функции в ряд. Сравнение численных значений функций позволяет судить о точности разложения. Более того, можно установить область значений  $x$ , в которой погрешность степенного ряда будет не более допустимой (при данном числе членов ряда).

*Пример 4.2.* Необходимо оценить погрешность ряда, полученного в результате разложения функции  $e^x$  в области  $x = [1 \div 2]$  при числе членов разложения  $n = 3, 4, 5$  (за начальное значение принять  $x_0 = 0$ ).

Технология решения задачи заключается в следующем.

□ Разложение  $e^x$  в ряд с помощью функции `TAYLOR(x, ex, 0, n)`.  
В результате разложения получим:

- $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!}$ , при  $n = 3$ ;
- $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!}$ , при  $n = 4$ ;

- $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!}$ , при  $n = 5$ .

### Примечание

Пусть эти функции находятся в строках #2, #3, #4.

- Табулирование  $e^x$  и функций, находящихся в строках #2, #3, #4. Функция табуляции в данном случае будет иметь вид:

$\text{VECTOR}([x, \#1, \#2, \#3, \#4], x, x_n, x_k, h)$ . Пусть функция  $e^x$  находится в строке #1, причем  $x = [1 \div 2]$ , а шаг  $h = 0,2$ . Тогда функция  $\text{VECTOR}$  будет иметь вид:  $\text{VECTOR}([x, \#1, \#2, \#3, \#4], x, 0, 2, 0.2)$ .

Процедуры вычисления и результаты табулирования показаны на рис. 4.12—4.14.

#1:  $\hat{e}^x$

#2:  $\text{TAYLOR}(\hat{e}^x, x, 0, 3)$

#3: 
$$\frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

#4:  $\text{TAYLOR}(\hat{e}^x, x, 0, 4)$

#5: 
$$\frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

#6:  $\text{TAYLOR}(\hat{e}^x, x, 0, 5)$

#7: 
$$\frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1$$

**Рис. 4.12.** Разложение  $e^x$  в ряд с помощью функции  $\text{TAYLOR}$

#8: 
$$\text{VECTOR}\left(\left[x, \hat{e}^x, \frac{x^3}{6} + \frac{x^2}{2} + x + 1, \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1, \frac{x^5}{120} + \frac{x^4}{24} + \frac{x^3}{6} + \frac{x^2}{2} + x + 1\right], x, 0, 2, 0.2\right)$$

**Рис. 4.13.** Табулирование функций

	0	1	1	1	1
#9:	0.2	1.221402758	1.221333333	1.2214	1.221402666
	0.4	1.491824697	1.490666666	1.491733333	1.491818666
	0.6	1.822118800	1.816	1.8214	1.822048
	0.8	2.225540928	2.205333333	2.2224	2.225130666
	1	2.718281828	2.666666666	2.708333333	2.716666666
	1.2	3.320116922	3.208	3.2944	3.315136
	1.4	4.055199966	3.837333333	3.9974	4.042218666
	1.6	4.953032424	4.562666666	4.835733333	4.923114666
	1.8	6.049647464	5.392	5.8294	5.986864
	2	7.389056098	6.333333333	7	7.266666666

**Рис. 4.14.** Оценка ряда, полученного  
в результате разложения функции  $e^x$

Из решения видно, что погрешности рядов при числе членов  $n = 3, 4, 5$  в диапазоне  $x = [1 \div 2]$  достаточно большие. Они уменьшаются с увеличением  $n$ . Можно показать, что для вычисления значения  $e^x$ , с точностью до шести знаков, необходимо иметь ряд Тейлора с числом членов разложения равным 9.

## Способ 2. Визуализация решения

Сущность этого способа состоит в следующем.

Исходная функция  $f(x)$  и степенной ряд, ей соответствующий, представляются в виде графика. Сравнение графиков позволяет судить о точности степенного ряда и допустимом диапазоне значений аргумента.

Технология использования данного способа состоит в следующем:

- разложение  $f(x)$  в ряд Тейлора;
- выделение на экране функции  $f(x)$ ;
- построение графика функции  $f(x)$  с помощью команд **2D-plot window | Plot Expression**;
- возврат в математическое окно путем нажатия кнопки **Algebra window**;

- построение графика функции, полученной в результате ее разложения функции;
- сравнение графиков.

При желании можно построить график разности функций, который будет характеризовать погрешность ряда Тейлора.

На рис. 4.15 приведены графики функций  $e^x$  и ряда Тейлора с числом членов  $n = 5$ , построенные по данным предыдущего примера.

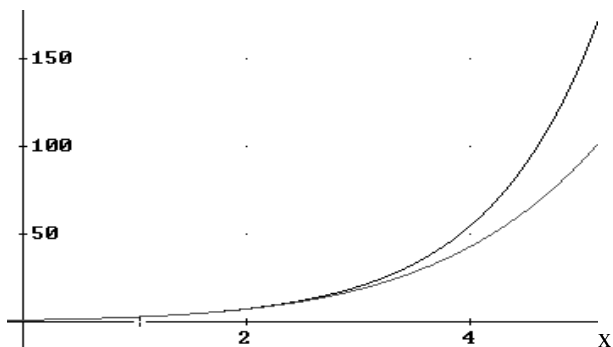


Рис. 4.15. Графики функций  $e^x$  и ряда Тейлора при  $n = 5$

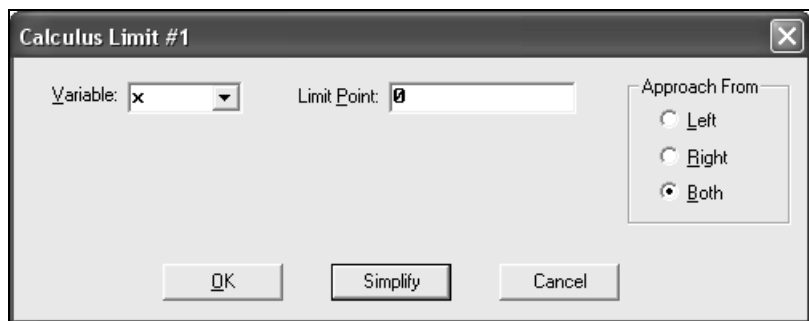
## 4.3. Вычисление пределов

Пределы в системе Derive 5 вычисляются с использованием кнопки — функции **Lim**, находящейся в пункте **Calculus** главного меню.

Технология вычисления пределов состоит в следующем:

- ввод выражения, предел которого необходимо найти;
- выполнение команды **Calculus | Limit** или щелчок мышью по кнопке **Lim** на панели инструментов (на экране формируется окно **Calculus Limit**, показанное на рис. 4.16);
- установка имени переменной в строке ввода **Variable**, а ее предельного значения — в строке ввода **Limit Point**;

- ☐ на панели переключателей **Approach From** устанавливается флажок нужной области изменения переменной:
  - **Left** — слева;
  - **Right** — справа;
  - **Both** — активизируется вставка;
- ☐ выполнение команды **Simplify** или **Approximate** (на экране формируется ответ).



**Рис. 4.16.** Окно установки параметров вычисления пределов

$$\begin{array}{ll}
 \#1: & \frac{\sin(x)}{x} \\
 \#2: & \lim_{x \rightarrow 0} \frac{\sin(x)}{x} \\
 \#3: & 1 \\
 \#4: & (1+x)^{1/x} \\
 \#5: & \lim_{x \rightarrow 0} (1+x)^{1/x} \\
 \#6: & e \\
 \#7: & \left( \frac{a^x + b^x}{2} \right)^{1/x} \\
 \#8: & \lim_{x \rightarrow 0} \left( \frac{a^x + b^x}{2} \right)^{1/x} \\
 \#9: & \sqrt{a} \cdot \sqrt{b}
 \end{array}$$

**Рис. 4.17.** Вычисление предела функции

Рассмотрим примеры вычисления следующих пределов:

☐  $\sin(x)/x$  при  $x \rightarrow 0$  ;

☐  $(1+x)^{\frac{1}{x}}$  при  $x \rightarrow 0$  ;

☐  $((a^x + b^x)/2)^{\frac{1}{x}}$  при  $x \rightarrow 0$  .

Вычислительные процедуры и результаты вычислений пределов указанных функций показаны на рис. 4.17.

## 4.4. Вычисление суммы ряда

Вычисление суммы ряда осуществляется в системе Derive с помощью команды **Sum**. Компьютерная технология суммирования членов ряда определяется последовательностью следующих действий:

- ☐ ввод функции суммирования;
- ☐ выполнение команды **Calculus | Sum** (на экране появляется окно **Calculus Sum**, показанное на рис. 4.18);

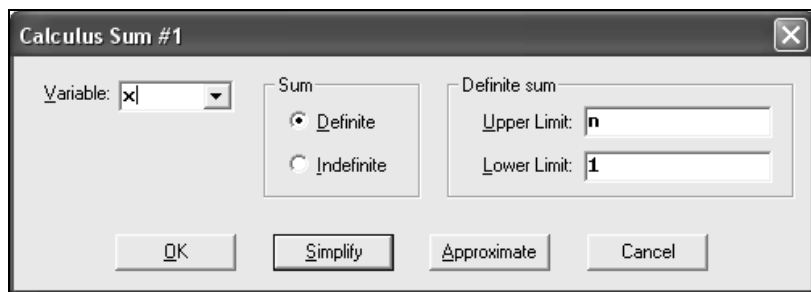


Рис. 4.18. Окно настройки элементов суммирования

- ☐ определение в строке ввода **Variable** имени индексной переменной;

- ❑ на панели **Sum** активизируется флажок **Definite**;
- ❑ на панели **Definite sum** (в соответствующих строках ввода) устанавливаются пределы суммирования;
- ❑ выполнение команды **Simplify** или **Approximate** (на экране формируется ответ, соответственно, в символьной или числовой форме).

Суммирование элементов вектора в системе Derive 5 осуществляется по командам **Element** и **Sum**. В этом случае применяется следующая последовательность действий:

1. Образование вектора элементов суммирования.
2. Ввод команды **Element** ( $\#1, n$ ), где  $\#1$  — номер строки вектора (в рассматриваемом случае строка  $\#1$ ),  $n$ -переменная суммирования.
3. Выполнение команды **Calculus | Sum** или щелчок мыши по кнопке  $\Sigma$ , расположенной на панели инструментов (на экране формируется окно — **Calculus Sum**).
4. Установка переменной в строке ввода **Variable** (в нашем случае  $n$ ).
5. На панели **Sum** активизируется флажок **Definite**.
6. На панели **Definite sum** устанавливаются пределы суммирования:
  - **Lower Limit** — нижний;
  - **Upper Limit** — верхний.
7. Щелкнуть мышью по кнопке **Simplify** или **Approximate** (на экране формируется ответ — значение суммы чисел, являющихся элементами вектора суммирования).

Ниже приводятся примеры вычисления суммы символьных переменных, числовых рядов и элементов вектора (рис. 4.19). Вычисления выполнены по описанной выше технологии.

$$\begin{aligned}
 \#1: & k^2 \\
 \#2: & \sum_{k=1}^n k^2 \\
 \#3: & \frac{n \bullet (n+1) \bullet (2n+1)}{6} \\
 \#4: & \frac{1}{k} \\
 \#5: & \sum_{k=1}^{100} \frac{1}{k} \\
 \#6: & 5.187377517 \\
 \#7: & [1, 2, 5, 4, 7, 6, 8, 9, 3, 1] \\
 \#8: & ELEMENT([1, 2, 5, 4, 7, 6, 8, 9, 3, 1], n) \\
 \#9: & \sum_{n=2}^8 ELEMENT([1, 2, 5, 4, 7, 6, 8, 9, 3, 1], n) \\
 \#10: & 41
 \end{aligned}$$

**Рис. 4.19.** Примеры вычисления суммы символьных переменных, числовых рядов и элементов вектора

$$\begin{aligned}
 \#1: & i! \\
 \#2: & \prod_{i=1}^5 i! \\
 \#3: & 34560 \\
 \#4: & 1 - \frac{1}{(2k+1)^2} \\
 \#5: & \prod_{k=1}^{\infty} \left( 1 - \frac{1}{(2k+1)^2} \right) \\
 \#6: & \frac{\pi}{4} \\
 \#7: & [2, 5, 7, -3, 8, -5, 12, -7, 15, -21] \\
 \#8: & ELEMENT([2, 5, 7, -3, 8, -5, 12, -7, 15, -21], n) \\
 \#9: & \prod_{n=1}^{10} ELEMENT([2, 5, 7, -3, 8, -5, 12, -7, 15, -21], n) \\
 \#10: & 222264000
 \end{aligned}$$

**Рис. 4.20.** Вычисление произведения ряда



## 4.5. Вычисление произведения ряда

Вычисление произведения ряда осуществляется командой **Produkt**. Технология вычисления произведения отличается от технологии вычисления суммы ряда лишь тем, что вместо знака  $\sum$  на панели инструментов используется знак произведения  $\Pi$ .

Ниже приводятся примеры вычисления произведения ряда (рис. 4.20).

## 4.6. Вычисление производных

Вычисление производной любого порядка в системе Derive 5 осуществляется с помощью команды **Differentiate**. Технология вычисления производной в системе Derive:

- ☐ ввод функции  $f(x)$ ;
- ☐ выполнение команды **Calculus | Differentiate** или щелкнуть мышью по кнопке  $\partial$ , расположенной на панели инструментов (на экране, в этом случае, формируется окно **Calculus Differentiate**, которое показано на рис. 4.21);
- ☐ ввод переменной дифференцирования в строке ввода — **Variable**;
- ☐ ввод порядка производной в строке ввода — **Order**;
- ☐ выполнение команды **Simplify** (на экране формируется иско-мая производная).

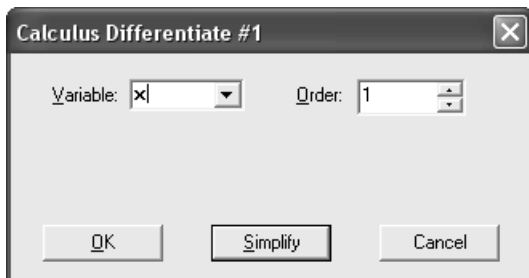


Рис. 4.21. Окно установки параметров дифференцирования

На рис. 4.22 показаны примеры вычисления производных в соответствии с описанной выше технологией.

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

$$\frac{d}{dx} \frac{x - 1}{x + 1}$$

$$\left(\frac{d}{dx}\right)^7 \frac{x - 1}{x + 1}$$

$$x^!$$

$$\frac{d}{dx} x^!$$

$$a^x \cdot b^x$$

$$\frac{d}{dx} (a^x \cdot b^x)$$

$$\frac{2}{(x + 1)^2}$$

$$\frac{10080}{(x + 1)^8}$$

$$\frac{d}{dx} x^!$$

$$a^x \cdot b^x \cdot (\ln(a) + \ln(b))$$

**Рис. 4.22.** Примеры вычисления производных



## Алгебра матриц и векторов

### 5.1. Представление векторов и матриц на экране ПК

Вектор и матрицу в системе Derive 5 можно отобразить на экране монитора и ввести в память ПК несколькими способами.

#### 5.1.1. Набор вектора (матрицы) с клавиатуры

Вектор или матрица представляются в строке ввода в следующем виде:

- $[x_1, x_2, \dots, x_n]$  — вектор с числом элементов, равным  $n$ ;
- $[[a_1, a_2, a_3], [b_1, b_2, b_3], [c_1, c_2, c_3]]$  — матрица размером  $3 \times 3$ .

Элементами вектора или матрицы могут быть символьные или числовые переменные (положительные или отрицательные, вещественные или комплексные). Система позволяет в качестве элементов вектора или матрицы использовать математические выражения. На рис. 5.1 и 5.2 показаны примеры представления векторов и матриц, элементами которых являются числовые и символьные значения.

В строках #1 и #5 показаны вектор с числом элементов 9 и матрица размером  $4 \times 3$ . Элементами вектора и матрицы являются вещественные числа. В строках #2 и #6 изображены вектор и матрица размером  $3 \times 3$  с элементами символьных переменных.

В строках #3 и #7 находятся вектор и матрица  $3 \times 3$ , элементами которых являются целые и комплексные числа. В строках #4 и #8 находятся вектор и матрица  $3 \times 3$  с элементами, содержащими функции.

$$\begin{aligned}
 \#1: & [1, 3, 5, 6, 8.5, 12, 14, 19, 23] \\
 \#2: & [a1, a2, a3, b1, b2, c3] \\
 \#3: & [5, 2 + 3 \cdot i, 3 - 2 \cdot i, 6, 7 \cdot i + 4, 3 - 12 \cdot i] \\
 \#4: & \left[ \hat{e}^x + 1, \sin(x) - 3 \cdot x + 2, \ln\left(\frac{x}{2}\right), \cos(x + 3), \hat{e}^x + \ln(x) - x - 1 \right] \\
 \#5: & \begin{bmatrix} 5 & 3 & 2 \\ 4 & 5 & 3 \\ 1 & 0 & 0 \\ 1 & 7 & 4 \end{bmatrix} \\
 \#6: & \begin{bmatrix} a1 & a2 & a3 \\ b1 & b2 & b3 \\ c1 & c2 & c3 \end{bmatrix} \\
 \#7: & \begin{bmatrix} 1 & 2 \cdot i + 3 & 5 \cdot i - 1 \\ i - 3 & 6 & 9 \\ 3 & 5 \cdot i + 3 & 0 \end{bmatrix} \\
 \#8: & \begin{bmatrix} \cos(x) & \hat{e}^x + 1 & 4 \\ 3 & \sin(x) & \ln(x) - x^2 + 1 \\ \sqrt{x} + 7 & \frac{x - 1}{x + 1} & 0 \end{bmatrix}
 \end{aligned}$$

**Рис. 5.1.** Представление векторов и матриц с элементами, являющимися комплексными числами и функциями

## 5.1.2. Использование пункта главного меню *Author*

Щелкая левой кнопкой мыши по позиции главного меню **Author**, а затем по позиции **VECTOR** или **MATRIX**, получим на экране окно ввода размера матрицы или вектора (рис. 5.2 и 5.3).

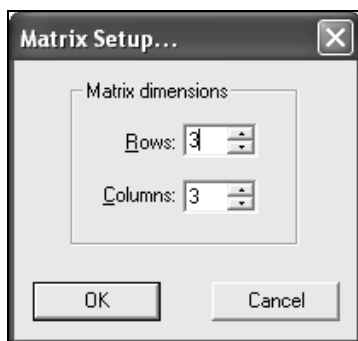


Рис. 5.2. Окно установки размера матрицы

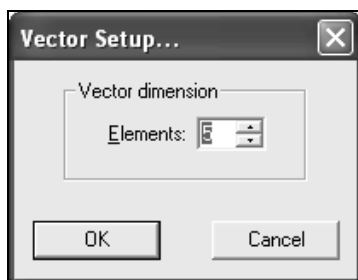


Рис. 5.3. Окно установки размера вектора

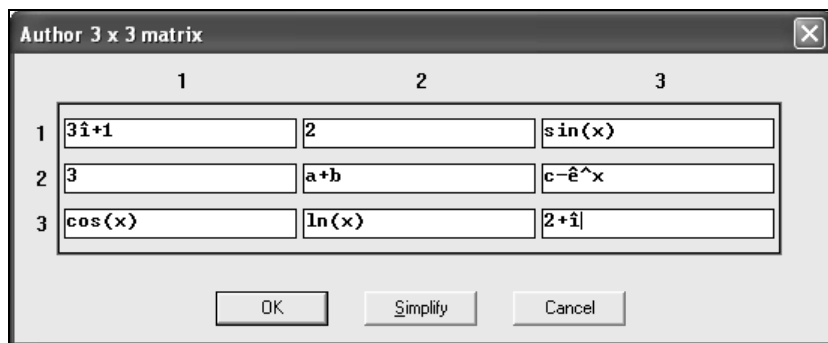


Рис. 5.4. Окно ввода элементов матрицы

При вводе задается число элементов (**Elements**) вектора или число строк (**Rows**) и число столбцов (**Columns**) матрицы. Нажатием

кнопки **ОК** выводится пустое окно для ввода значений элементов вектора или матрицы. Элементами, в данном случае, могут быть как действительные и комплексные числа, так и символьные переменные и даже математические выражения. На рис. 5.4 показано окно ввода элементов матрицы размером  $3 \times 3$  со смешанными элементами.

После нажатия кнопки **ОК** или клавиши  $\langle \text{Enter} \rangle$  на экране появляется вектор или матрица. При использовании этого способа можно не обращаться к меню **Author**, а воспользоваться кнопкой **Вектор** или **Матрица**, расположенных на панели инструментов. При нажатии кнопки появится окно ввода размера вектора или матрицы.

## 5.2. Операции с векторами

Система Derive 5 допускает следующие действия с векторами:

- ☐ умножение (деление) вектора на целое или дробное, положительное или отрицательное число (результатом операции будет вектор, каждый элемент которого умножен или разделен на число);
- ☐ сложение (вычитание) вектора с целым или дробным, положительным или отрицательным числом (результатом операции будет вектор, каждый элемент которого является суммой или разностью элемента исходного вектора и числа);
- ☐ сложение, вычитание, умножение и деление векторов;
- ☐ возведение вектора в степень.

Система Derive 5 допускает выполнение перечисленных операций в случаях, когда элементами вектора являются символьные и комплексные значения, а также любые математические выражения.

*Пример 5.1.* Пусть даны два вектора:  $[1, 3, 2, 7]$  и  $[2, 1, 3, 4]$ . Необходимо выполнить следующие операции:

- ☐ первый вектор умножить и разделить на число 5 и символьную переменную;

- ☐ возвести первый вектор в степени 2,3,4,-2,-3,-4;
- ☐ выполнить операции сложения, вычитания, умножения и деления векторов.

При вводе векторов и математических операций используются те же символы, что и в случае операций над числами и переменными (+, -, \*, /, ^). При этом операция умножения также будет выполняться в том случае, если вместо знака \* используется точка или пробел.

В результате вычислений получим решение, которое на экране монитора будет иметь вид:

$$\#1: [1, 3, 2, 7]$$

$$\#2: [1, 3, 2, 7].5$$

$$\#3: [5, 15, 10, 35]$$

$$\#4: [1, 3, 2, 7].a$$

$$\#5: [a, 3.a, 2.a, 7.a]$$

$$\#6: \frac{[1,3,2,7]}{5}$$

$$\#7: \left[ \frac{1}{5}, \frac{3}{5}, \frac{2}{5}, \frac{7}{5} \right]$$

$$\#8: [0.2, 0.6, 0.4, 1.4]$$

$$\#9: [1, 3, 2, 7]^2$$

$$\#10: 63$$

$$\#11: [1, 3, 2, 7]^3$$

$$\#12: [63, 189, 126, 441]$$

$$\#13: [1, 3, 2, 7]^4$$

$$\#14: 3969$$

$$\#15: [1, 3, 2, 7]^{-2}$$

$$\#16: \frac{1}{63}$$

$$\#17: [1, 3, 2, 7]^{-3}$$

$$\#18: \left[ \frac{1}{3969}, \frac{1}{1323}, \frac{1}{3969}, \frac{1}{567} \right]$$

$$\#19: \begin{bmatrix} 0.00025195263, & 0.00075585789, & 0.00050390526, \\ 0.0017636684 \end{bmatrix}$$

$$\#20: [1, 3, 2, 7]^{-4}$$

$$\#21: \frac{1}{3969}$$

$$\#22: [1, 3, 2, 7] + [2, 1, 3, 4]$$

$$\#23: [3, 4, 5, 11]$$

$$\#24: [1, 3, 2, 7] - [2, 1, 3, 4]$$

$$\#25: [-1, 2, -1, 3]$$

$$\#26: [1, 3, 2, 7] \cdot [2, 1, 3, 4]$$

$$\#27: 39$$

$$\#28: \frac{[1, 3, 2, 7]}{[2, 1, 3, 4]}$$

$$\#29: \frac{13}{10}$$

## 5.3. Виды и характеристики матриц

Существует большое число различных видов матриц. Ниже приводятся основные из них.

- Прямоугольная ( $M$ ) — таблица, содержащая  $m$  строк и  $n$  столбцов элементов.
- Квадратная ( $A$ ) — матрица, у которой число строк равно числу столбцов.
- Транспонированная ( $A^T$ ) — матрица, у которой строки и столбцы меняются местами.
- Единичная ( $E$ ) — квадратная матрица, у которой диагональные элементы равны 1, а остальные равны 0.
- Сингулярная ( $C$ ) — квадратная матрица, детерминант которой равен нулю.
- Обратная ( $M^{-1}$ ) — матрица, произведение которой на исходную квадратную матрицу образует единичную матрицу, т. е.  $AM^{-1} = E$ .



- Идемпотентная ( $P$ ) — матрица, удовлетворяющая условию  $P^2 = P$ .
- Симметричная ( $S$ ) — матрица, удовлетворяющая условию  $A^T = A$ .
- Кососимметричная ( $KS$ ) — матрица, удовлетворяющая условию  $A^T = -A$ .
- Ортогональная — матрица, удовлетворяющая условию  $A^T = A^{-1}$ .
- Инволютивная ( $I$ ) — матрица, удовлетворяющая условию  $I^2 = E$ .

Некоторые из перечисленных видов матриц могут быть получены из исходной матрицы путем ее преобразования с помощью функций системы Derive 5.

Единичную матрицу  $n \times n$  можно получить с помощью функции `IDENTITY_MATRIX (n)`.

Транспонированная матрица из исходной  $A$  образуется при вводе символа  $a^T$  и команды **Simplify** или **Approximate** на панели инструментов. В этих преобразованиях  $a^T$  — имя матрицы  $A$ . Вместо символа  $a^T$  можно ввести номер строки экрана, в которой находится матрица  $A$ .

Обратная матрица образуется при вводе символа  $a^{-1}$  и команды **Simplify** или **Approximate**.

*Пример 5.2.* Пусть исходной является матрица:

$$a = \begin{vmatrix} 1 & 3 & 7 & 4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}.$$

Необходимо с помощью указанных выше функций образовать единичную, транспонированную и обратную матрицы. Результаты преобразования исходной матрицы, на экране монитора, показаны на рис. 5.5.

$$\#1: \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}$$

#2: IDENTITY\_MATRIX (4)

$$\#3: \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\#4: \mathbf{a} = \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}$$

#5:  $\mathbf{a}'$

$$\#6: \begin{vmatrix} 1 & 0 & 6 & 4 \\ 3 & -2 & -3 & -7 \\ 7 & 4 & 1 & 3 \\ -4 & 5 & 2 & 1 \end{vmatrix}$$

$$\#7: \mathbf{a} = \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix},$$

$$\#8: \begin{vmatrix} 1 & 0 & 6 & 4 \\ 3 & -2 & -3 & -7 \\ 7 & 4 & 1 & 3 \\ -4 & 5 & 2 & 1 \end{vmatrix}$$

#9:  $\mathbf{a}^{-1}$

$$\#10: \begin{vmatrix} \frac{11}{431} & \frac{39}{862} & \frac{77}{862} & -\frac{23}{431} \\ \frac{21}{431} & \frac{47}{1724} & \frac{137}{1724} & \frac{127}{862} \\ \frac{88}{431} & \frac{193}{1724} & -\frac{61}{1724} & \frac{63}{862} \\ \frac{27}{431} & -\frac{309}{1724} & \frac{53}{1724} & -\frac{83}{862} \end{vmatrix}$$

$$\#11: \begin{vmatrix} 0.02552204176 & 0.04524361948 & 0.08932714617 & -0.05336426914 \\ 0.04872389791 & 0.02726218097 & 0.07946635730 & 0.1473317865 \\ 0.2041763341 & 0.1119489559 & 0.03538283962 & 0.07308564686 \\ -0.062645011 & -0.1792343387 & 0.03074245939 & -0.09628770301 \end{vmatrix}$$

Рис. 5.5. Варианты преобразования матрицы

**Примечание**

При образовании транспонированной матрицы обязательным действием является присвоение ей имени (например, "a" или другой переменной) с помощью знака присвоения ( $:=$ ). Присвоение матрице имени можно осуществить путем ввода следующего выражения:  $a := \#1$ , где #1 — номер строки, в которой находится матрица (первая строка). В этом случае переменная  $a$  не будет выводиться на экран (см. рис. 5.6, выражение #7).

### 5.3.1. Характеристики матриц

Основными характеристиками матриц являются следующие.

- ☐ *Ранг матрицы* — наибольший порядок определителя матрицы.
- ☐ *След матрицы* — сумма диагональных элементов матрицы (вычисляется с помощью функции  $\text{TRACE}(A)$ ).
- ☐ *Значение детерминанта* — значение определителя матрицы (вычисляется с помощью функции  $\text{DET}(A)$ ).
- ☐ *Собственное значение матрицы* — корни характеристического полинома (вычисляется с помощью функции  $\text{EIGENVALUES}(A, x)$ , где  $A$  — квадратная матрица,  $x$  — имя результата вычисления).
- ☐ *Характеристический полином* — определитель вида  $A - xE$ , где  $A$  — квадратная матрица,  $E$  — единичная матрица,  $x$  — аргумент многочлена (вычисляется с помощью функции  $\text{CHARPOLY}(A, x)$ , где  $A$  — матрица,  $x$  — аргумент многочлена).

Для выделения элемента матрицы используется функция:

$\text{Element}(A, m, n)$ ,

где  $A$  — матрица,  $m$  — номер строки,  $n$  — номер столбца.

Вычислим основные характеристики матрицы:

- ☐ след;
- ☐ значение детерминанта;
- ☐ собственное значение;
- ☐ характеристический полином.

В качестве примера возьмем матрицу размером  $4 \times 4$  из предыдущего примера.

Процедуры решения задачи очевидны, поэтому приведем лишь конечные результаты. Ответы на экране монитора будут иметь следующий вид:

```
#1:       $\begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}$ 

#2:      TRACE       $\begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}$ 

#3:                                             1

#4:      DET       $\begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}$ 

#5:                                             -1724

#6:      EIGENVALUES  $\left( \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}, x \right)$ 

#7:      [x=7.0.23738296, x=-4.992768654, x=-0.5154848210+
        6.992570618.i, x=-0.5154848210-6.992570618 i ]

#8      CHARPOLY  $\left( \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}, x \right)$ 

#9      x^4-x^3+12.x^2-136.x-1724

#10     ELEMENT  $\left( \begin{vmatrix} 1 & 3 & 7 & -4 \\ 0 & -2 & 4 & 5 \\ 6 & -3 & 1 & 2 \\ 4 & -7 & 3 & 1 \end{vmatrix}, 2, 3 \right)$ 

#11:
```

Система Derive 5 позволяет выполнять матричные операции точно так же, как векторные. Матричными операциями являются:

- ☐ умножение (деление) на постоянный множитель;
- ☐ возведение в степень матрицы;
- ☐ сложение матриц;
- ☐ вычитание матриц;
- ☐ умножение матриц;
- ☐ деление матриц.

При этом математические операторы перечисленных действий те же, что в обычной алгебре. Вычисления рассмотрим на примере.

*Пример 5.3.*

Даны следующие две матрицы:

$$A_1 = \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix}; A_2 = \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 1 & 5 \end{vmatrix}.$$

Требуется:

- ☐ матрицу  $A_1$  умножить и разделить на 5;
- ☐ матрицу  $A_2$  возвести в квадрат;
- ☐ выполнить операции:
  - $A_1 + A_2$ ;
  - $A_1 - A_2$ ;
  - $A_1 \times A_2$ ;
  - $\frac{A_1}{A_2}$ .

Выполняя эти операции аналогично векторным, будут получены результаты, показанные на рис. 5.6.

Алгебра векторов и матриц широко используется в расчетах. Она позволяет представлять математические функции в виде таблиц, решать системы уравнений, вычислять погрешности математических моделей и многое другое. В гл. 7 будут рассмотрены методы решения систем уравнений с помощью матриц.

$$\begin{array}{ll}
 \#1: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} \\
 \#2: & \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 3 & 4 \end{vmatrix} \\
 \#3: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} \cdot 5 \\
 \#4: & \begin{vmatrix} 5 & 10 & 15 \\ 15 & 20 & 5 \\ 25 & 15 & 20 \end{vmatrix} \\
 \#5: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} \\
 \#6: & \begin{vmatrix} 1 & 2 & 3 \\ 5 & 5 & 5 \\ 3 & 4 & 1 \\ 5 & 5 & 5 \\ 1 & 5 & 5 \end{vmatrix} \\
 \#7: & \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 1 & 5 \end{vmatrix}^2 \\
 \#8: & \begin{vmatrix} 16 & 15 & 20 \\ 37 & 24 & 55 \\ 41 & 19 & 36 \end{vmatrix} \\
 \#9: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} + \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 1 & 5 \end{vmatrix} \\
 \#10: & \begin{vmatrix} 4 & 4 & 4 \\ 4 & 8 & 7 \\ 10 & 4 & 9 \end{vmatrix} \\
 \#11: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} - \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 1 & 5 \end{vmatrix} \\
 \#12: & \begin{vmatrix} -2 & 0 & 2 \\ 2 & 0 & -5 \\ 0 & 2 & -1 \end{vmatrix} \\
 \#13: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} \cdot \begin{vmatrix} 3 & 2 & 1 \\ 1 & 4 & 6 \\ 5 & 1 & 5 \end{vmatrix} \\
 \#14: & \begin{vmatrix} 20 & 13 & 28 \\ 18 & 23 & 32 \\ 30 & 26 & 43 \end{vmatrix} \\
 \#15: & \begin{vmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 5 & 3 & 4 \end{vmatrix} \\
 \#16: & \begin{vmatrix} 7 & 32 & 4 \\ 73 & 73 & 73 \\ 123 & 20 & 29 \\ 73 & 73 & 73 \end{vmatrix}
 \end{array}$$

Рис. 5.6. Действия над матрицами



## Решение алгебраических и трансцендентных уравнений

### 6.1. Решение уравнений с одним неизвестным

Решение алгебраических и трансцендентных уравнений осуществляется с помощью функции `SOLVE`. Эта функция представляется в одном из следующих видов:

- ☐ `Solve(f(x), x);`
- ☐ `Solve(f(x), x, Real);`
- ☐ `Nsolve(f(x), x);`
- ☐ `Nsolve(f(x), x, Real);`
- ☐ `Approxsolve(f(x), x);`
- ☐ `Approxsolve(f(x), x, Real);`
- ☐ `Nsolve(f(x), x, a, b);`

В функциях приняты следующие обозначения:

- ☐  $f(x)$  — алгебраическое или трансцендентное уравнение;
- ☐  $x$  — неизвестное;
- ☐  $[a, b]$  — область изоляции корня.

Компьютерная технология определения корней с помощью приведенных функций исключительно проста и состоит в следующем:

- ☐ ввод функции (на экране формируется функция, номер строки которой обозначается символом  $\#k$ );

- ❑ щелчок мыши по кнопке **Simplify**, расположенной на панели инструментов (на экране формируется ответ — значения корней уравнения).

Однако эта технология удобна не во всех случаях. Например, если уравнение сложное и приходится применять скобки. Кроме того, набор функции занимает много времени, особенно у неопытного пользователя, когда приходится многократно редактировать функцию из-за орфографических или синтаксических ошибок.

Система Derive 5 освобождает от необходимости набора функций. Она требует набора только уравнения.

Наиболее рациональной является следующая технология определения корней уравнения.

- ❑ Ввод уравнения  $f(x) = 0$ , при этом оно может быть представлено в любой форме, например:

$$ax^2 + bx - 1 = 0, \quad ax^2 + bx - 1, \quad ax^2 - 1 = -bx,$$

$$ax^2 + bx = 1, \quad x = \frac{1 - ax^2}{b}.$$

- ❑ Выбор вида функции **SOLVE**.
- ❑ Выполнение команды **Solve | Expression** или щелчок мышью по кнопке **Solve Expression** на панели инструментов.

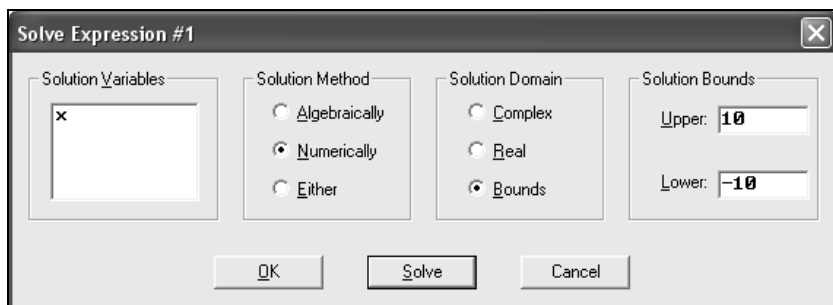
### Примечание

На экране формируется окно **Solve Expression** с четырьмя панелями: **Solution Variables**, **Solution Method**, **Solution Domain**, **Solution Bounds** (рис. 6.1).

На панели **Solution Variables** отражены все переменные уравнения.

Пользователь должен отметить (путем засветки) искомое неизвестное. Остальные панели предназначены для образования перечисленных выше функций, путем установки значений соответствующих флажков. Каждой функции соответствует определенная комбинация значений флажков (табл. 6.1).





**Рис. 6.1.** Окно выбора метода решения уравнения и ввода данных **Solve Expression**

**Таблица 6.1.** Соответствие значений флажков и функций

Комбинация установленных флажков	Функция	Содержание функции
<b>Algebraically и Complex</b>	<code>Solve(f(x), x)</code>	Решение уравнений в аналитическом виде
<b>Algebraically и Real</b>	<code>Solve(f(x), x, Real)</code>	Определение только действительных корней
<b>Numerically и Complex</b>	<code>Nsolve(f(x), x)</code>	Определение корней в численном виде
<b>Numerically и Real</b>	<code>Nsolve(f(x), x, Real)</code>	Определение действительных корней в численном виде
<b>Numerically и Bounds и Lower-a, Upper-b</b>	<code>Nsolve(f(x), x, a, b)</code>	Определение численного значения корня из области изоляции $[a, b]$
<b>Either и Complex</b>	<code>Approx(Solve(f(x), x))</code>	Аналогично команде <code>Nsolve(f(x), x)</code>
<b>Either и Real</b>	<code>Approx(Solve(f(x), x, Real))</code>	Аналогично команде <code>Nsolve(f(x), x, Real)</code>

### Примечание

Откликом функции `Nsolve(f(x), x, a, b)` будет лишь один корень из области  $[a, b]$ , хотя в этой области может быть несколько корней. Если в этой области  $[a, b]$  корней нет, то `Derive 5` повторит функцию.

#### □ Получение решения.

После выбора функции нажимается одна из кнопок **OK** или **Solve**, расположенных в нижней части окна **Solve Expression**. При нажатии кнопки **Solve** на экране формируется ответ — значение искомого корня (корней). При нажатии кнопки **OK** на экране формируется функция, которую нужно исполнить нажатием кнопки **Simplify**, расположенной на панели инструментов.

*Пример 6.1.* Пусть необходимо решить уравнение  $(x^4 - 1)(x^2 - 2a + 1) = 0$ . Решение заключается в выполнении следующих действий:

- набор и ввод уравнения — на экране формируется уравнение в виде #1:  $(x^4 - 1)(x^2 - 2a + 1)$ ;
- щелчок мыши по кнопке **Solve Expression** — на экране формируется окно **Solve Expression**;
- засветка  $x$  на панели **Solution Variables** и установка флажков **Algebraically** и **Complex**;
- выполнение команды **Solve**.

На экране формируется ответ:

$$x = -i, x = i, x = -\sqrt{2a-1}, x = \sqrt{2a-1}, x = -1, x = 1.$$

Программа нашла все шесть корней уравнения. При реализации функции `Solve(f(x), x, Real)` путем установки флажков **Algebraically** и **Real** в ответе будут только действительные корни:

$$x = -1, x = 1, x = -\sqrt{2a-1}, x = \sqrt{2a-1}.$$

## 6.2. Решение уравнений в аналитическом виде

Система Derive 5 позволяет решать уравнение с одним неизвестным, получая при этом ответы в аналитическом или численном виде. Возможность решения уравнений в аналитическом виде является одним из признаков интеллектуальности данной системы.

Решение уравнения в аналитическом виде является наиболее полезным. Это объясняется обобщенностью такого решения. Ответ в этом случае получается в виде формулы, справедливой для широкого диапазона коэффициентов исходного уравнения.

Предположим, что необходимо решить следующее уравнение —  $\ln(ax + b) = c$ , в котором  $x$  является неизвестным, а переменные  $a, b, c$  могут принимать различные численные значения. Решение этого уравнения имеет вид:

$$x = \frac{10^c - b}{a}.$$

Теперь нет необходимости решать исходное уравнение каждый раз при новых значениях коэффициентов  $a, b, c$ . Достаточно их подставить в полученное решение и получить значение  $x$ .

Решение уравнения в аналитическом виде часто необходимо еще и потому, что над этим выражением необходимо производить математические операции.

Система Derive 5 позволяет получать решения в аналитическом виде с помощью следующих двух функций:  $\text{SOLVE}(f(x), x, \text{Real})$  и  $\text{SOLVE}(f(x), x)$ .

Первая функция выдает значения только действительных корней, вторая — действительных и комплексных.

Технологию вычисления этих функций рассмотрим на примерах.

*Пример 6.2.* Пусть необходимо решить следующее уравнение:

$$ax^2 + bx = 0.$$

Решение его всеми возможными способами показано на рис. 6.2.

#1:	$\text{SOLVE}(a \cdot x^2 + b \cdot x, x)$	
#2:		$x = -\frac{b}{a} \vee x = 0$
#3:	$\text{SOLVE}(a \cdot x^2 + b \cdot x, x, \text{Real})$	
#4:		$x = -\frac{b}{a} \vee x = 0$
#5:	$a \cdot x^2 + b \cdot x$	
#6:	$\text{SOLVE}(a \cdot x^2 + b \cdot x, x, \text{Real})$	
#7:		$x = -\frac{b}{a} \vee x = 0$
#8:	$\text{SOLVE}(a \cdot x^2 + b \cdot x, x)$	
#9:		$x = -\frac{b}{a} \vee x = 0$
#10:	$a \cdot x^2 + b \cdot x$	
#11:		$x = -\frac{b}{a} \vee x = 0$

**Рис. 6.2.** Решение уравнения  $ax^2 + bx = 0$   
всеми возможными способами

Рассмотрим способы, с помощью которых получено решение.

*Способ 1.*

- ☐ Ввод функции  $\text{SOLVE}(ax^2+bx, x)$ .
- ☐ Обращение к пункту главного меню **Simplify | Approximate** (на экране формируется окно **Approximate Expression**).
- ☐ На панели **Digits of Precision** предлагается выдать решение с десятью значащими цифрами. В нашем случае решение требуется получить в аналитическом виде, поэтому устанавливать точность решения не нужно. По этой причине можно не обра-

щаться к пункту меню **Simplify**, а щелкнуть мышью по кнопке **Simplify (=)** или **Approximate ( $\approx$ )** на панели инструментов.

- ☐ На экране формируется ответ в виде —  $x = \frac{b}{a}, x = 0$ .

### Примечание

В исходном уравнении коэффициенты  $a$  и  $b$  могут быть как вещественными, так и комплексными. Для программы они не различимы. Поэтому функция **SOLVE** может быть представлена в виде **SOLVE(ax2+bx, x, Real)**. При этом ответ будет прежним. Решения находятся в строках #1, #2, #3, #4.

### Способ 2.

- ☐ Ввод уравнения в произвольном виде, например,  $ax^2 + bx$ .
- ☐ Обращение к пункту главного меню **Solve | Expression** или нажатие кнопки **Solve Expression**, расположенной на панели инструментов, — на экране формируется окно **Solve Expression** с четырьмя панелями.
- ☐ Активизация на панели **Solution Variables** переменной  $x$ .
- ☐ Установка флажка **Algebraically** на панели **Solution Method**.
- ☐ Установка флажков — **Real** или **Complex** на панели **Solution Domain**.
- ☐ Нажатие кнопки **Solve** — на экране формируется ответ. При нажатии кнопки **OK** — на экране представляется функция **SOLVE(ax2+bx, x, Real)**, если был установлен флажок — **Real**, и функция **SOLVE(ax2+bx, x)**, если был установлен флажок **Complex**. Решение уравнения показано в строках #5, #6, #7, #8, #9 (см. рис. 6.2).

### Способ 3.

- ☐ Ввод уравнения в произвольном виде, например,  $ax^2 + bx$  (на рис. 6.2 уравнение определено в строке #10).
- ☐ Набор в окне ввода функции **SOLVE(#10, x)**. Не выводя набранную функцию на экран, щелкнуть мышью по кнопке

**Simplify** или **Approximate**, которые расположены слева от окна ввода функций.

Решение на экране определено в строках — #10, #11.

При наличии в уравнении комплексных корней функция **SOLVE** найдет решение в аналитическом виде только при условии, если будут определены области переменных, при которых имеются комплексные корни. В противном случае решение будет выдано программой в обычном символьном виде.

*Пример 6.3.* Пусть задано уравнение  $ax^2 + b = 0$ , при  $a > 0$  и  $b > 0$ . В этом случае:

$$x_{1,2} = \pm \sqrt{-\frac{b}{a}} = \pm i \sqrt{\frac{b}{a}}.$$

Если программе не сообщить область значений  $a$  и  $b$ , то решение будет:  $x_1 = -\frac{\sqrt{-b}}{\sqrt{a}}, x_2 = \frac{\sqrt{-b}}{\sqrt{a}}$ .

Если же программе сообщить, что  $a > 0$  и  $b > 0$ , то решение будет в следующем виде:

$$x = -\frac{i\sqrt{b}}{\sqrt{a}}, x = \frac{i\sqrt{b}}{\sqrt{a}}.$$

Диапазон изменения переменных устанавливается следующим образом:

- ☐ ввод уравнения;
- ☐ вызов функции **Declare | Variable Domain...** (на экране формируется окно **Declare Variable Domain**, показанное на рис. 6.3);
- ☐ ввод в строке ввода **Variable Name** имени переменной (в нашем случае переменной  $a$ );
- ☐ установка флажков — **Real** на панели **Domain**, **Positive (0, ∞)** на панели **Interval**;
- ☐ щелчок по кнопке **OK** (на экране формируется выражение вида —  $a: \in \text{Real}(0, \infty)$ , что означает положительное значение переменной  $a$  во всем диапазоне чисел от 0 до  $\infty$ ).

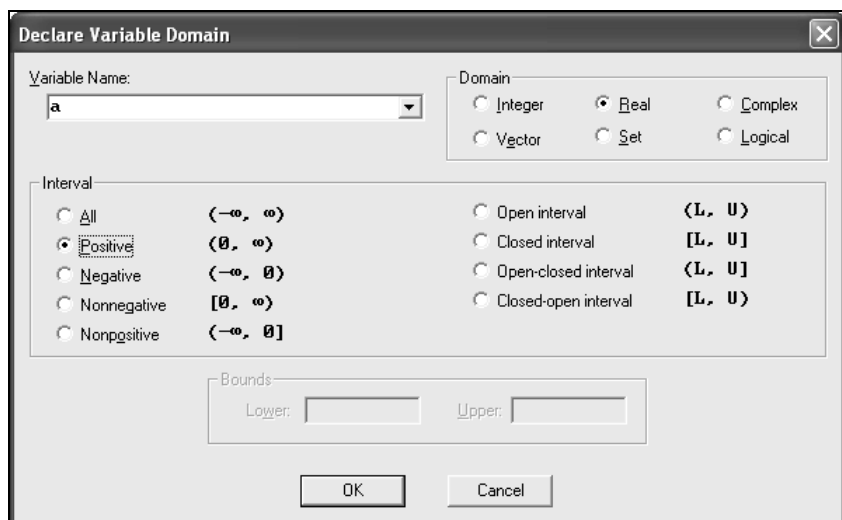


Рис. 6.3. Окно установки области изменения переменной

#1:  $a \cdot x^2 + b$

#2:  $\text{SOLVE}(a \cdot x^2 + b, x)$

#3: 
$$x = -\frac{\sqrt{-b}}{\sqrt{a}} \vee x = \frac{\sqrt{-b}}{\sqrt{a}}$$

#4:  $a \in \text{Real } (0, \infty)$

#5:  $b \in \text{Real } (0, \infty)$

#6:  $\text{SOLVE}(a \cdot x^2 + b, x)$

#7: 
$$x = -\frac{i \cdot \sqrt{b}}{\sqrt{a}} \vee x = \frac{i \cdot \sqrt{b}}{\sqrt{a}}$$

#8:  $\text{SOLVE}(a \cdot x^2 + b, x, \text{Real})$

#9:  $\text{false}$

Рис. 6.4. Решение уравнения  $ax^2 + bx$  с использованием областей изменения переменных

Аналогично присваивается положительное значение переменной  $b$ .

Теперь функция  $\text{SOLVE}(ax^2 + bx, x, \text{Complex})$  выдаст положительные значения корней, а функция  $\text{SOLVE}(ax^2 + bx, x, \text{Real})$  не будет реализована, ответ последует в виде значения — `false`. Решения, представляемые на экране, показаны на рис. 6.4.

Рассмотрим еще один пример.

*Пример 6.4.* Пусть необходимо решить уравнение  $x^5 - bx^3 - a^3x^2 + a^3b = 0$ . Его решение имеет вид (рис. 6.5):

```
#1:  x5 - b·x3 - a3·x2 + a3·b
#2:  SOLVE(x5 - b·x3 - a3·x2 + a3·b, x, Real)
#3:   $\left(x = -\frac{a}{2} \wedge a = 0\right) \vee x = a \vee x = -\sqrt{b} \vee x = \sqrt{b}$ 
#4:  SOLVE(x5 - b·x3 - a3·x2 + a3·b, x)
#5:   $x = -\frac{a}{2} - \frac{\sqrt{3} \cdot i \cdot a}{2} \vee x = -\frac{a}{2} + \frac{\sqrt{3} \cdot i \cdot a}{2} \vee x = a \vee x = -\sqrt{b} \vee x = \sqrt{b}$ 
#6:  b :∈ Real (-∞, 0)
#7:  SOLVE(x5 - b·x3 - a3·x2 + a3·b, x, Real)
#8:   $\left(x = -\frac{a}{2} \wedge a = 0\right) \vee x = a$ 
#9:  SOLVE(x5 - b·x3 - a3·x2 + a3·b, x)
#10:  $x = -\frac{a}{2} - \frac{\sqrt{3} \cdot i \cdot a}{2} \vee x = -\frac{a}{2} + \frac{\sqrt{3} \cdot i \cdot a}{2} \vee x = -i \cdot \sqrt{-b} \vee x = i \cdot \sqrt{-b} \vee x = a$ 
```

**Рис. 6.5.** Решение уравнения  $x^5 - bx^3 - a^3x^2 + a^3b = 0$

Если воспользоваться функцией  $\text{SOLVE}(x^5 - bx^3 - a^3x^2 + a^3b, x, \text{Real})$ , то программа выдаст решение в виде:

$$\left(x = -\frac{a}{2}, a = 0\right); x = a; x = -\sqrt{b}; x = \sqrt{b}.$$



Ответ в скобках означает, что комплексные корни не найдены. Программа нашла только вещественные корни. Если же воспользоваться функцией `SOLVE (x5 - bx3 - a3x2 + a3b, x, Complex)`, то будут определены все пять корней, два из которых комплексные:

$$x_1 = a; x_2 = -\sqrt{b}; x_3 = \sqrt{b};$$
$$x_4 = -\frac{a}{2} - \frac{\sqrt{3}}{2}ai; x_5 = -\frac{a}{2} + \frac{\sqrt{3}}{2}ai.$$

Решение находится в строках с #1 по #5 экрана.

В рассмотренном примере программе не нужно было сообщать область значений переменных  $a$  и  $b$ , если  $a$  имеет любой знак, а  $b > 0$ , то данное решение верно, однако пользователь должен иметь в виду, что корни  $x_2, x_3$  будут комплексными —  $x_{2,3} = \pm i\sqrt{b}$ .

Если установить отрицательную область значений  $b$ , то решение будет иметь вид, показанный в строках #6, #7, #8 (рис. 6.6). В данном случае программа нашла только один вещественный корень  $x = a$ , а при определении комплексных корней с помощью функции `SOLVE (f(x), x, Complex)` найдены все пять корней, из которых четыре комплексные. Решение находится в строках #9 и #10.

Из приведенных примеров следует, что при решении уравнений в аналитическом виде, во избежание ошибок, целесообразно задавать область изменения символьных переменных.

В табл. 6.2 приводятся примеры для самостоятельного решения задач определения корней алгебраических уравнений в аналитическом виде.

**Таблица 6.2.** Примеры задач решения алгебраических уравнений

№ п/п	Уравнения	Ответы
1	$x^3 - a = 0$	<p>a) <math>x = a^{1/3}</math>;</p> <p>b) <math>x_1 = a^{1/3}, x_2 = a^{1/3} \left( -\frac{1}{2} \pm \frac{\sqrt{3}}{2} i \right)</math>.</p>
2	$x^5 + (b-a)x^4 - abx^3 + ax^2 + ax(b-a) - a^2b = 0$	<p>a) <math>x_1 = a, x_2 = -a^{1/3}, x_3 = b</math>;</p> <p>b) <math>x_1 = a, x_2 = -a^{1/3}</math>,</p> <p>1. <math>x_3 = -b, x_{4,5} = a^{1/3} \left( \frac{1}{2} \pm \frac{\sqrt{3}}{2} i \right)</math>.</p>
3	$x^5 + bx^4 - a^4x - a^4b = 0$	<p>a) <math>x_1 = -a, x_2 = -a, x_3 = -b, (x = 0, a = 0)</math>;</p> <p>b) <math>x_1 = -a, x_2 = -a, x_3 = -b, x_{4,5} = \pm ai</math>.</p>
4	$x^4 + 2x^3(1+a) + x^2(4a-1) - 2x(1+a) - 4a = 0$	$x_1 = 1, x_2 = -1, x_3 = -2, x_4 = -2a$

Таблица 6.2 (продолжение)

№ п/п	Уравнения	Ответы
5	$x^5 - 3x^4 + (a + b^3)x^3 +$ $+ 3(b^3 - a)x^2 - ab^3x + 3ab^3 = 0$	<p>a) <math>x_1 = -\sqrt{-a}, x_2 = \sqrt{-a}, x_3 = 3;</math></p> <p>b) <math>x_1 = -\sqrt{-a}, x_2 = -\sqrt{-a}, x_3 = 3;</math></p> <p>2. <math>x_{4,5} = \pm i(-b)^{\frac{2}{3}}</math></p> <p>3. при <math>a &gt; 0, b &gt; 0</math></p> <p>a) <math>x = 3;</math></p> <p>b) <math>x_1 = 3, x_{2,3} = \pm i\sqrt{a}, x_{4,5} = i(-b)^{\frac{3}{2}}.</math></p>
6	$x^5 - ax^3 + 2x^2 - 2a = 0$	<p>a) <math>x_{1,2} = \pm\sqrt{a}, x_3 = -2^{\frac{1}{3}};</math></p> <p>b) <math>x_{1,2} = \pm\sqrt{a}, x_3 = -2^{\frac{1}{3}},</math></p> <p>4. <math>x_{4,5} = \frac{1}{2} \pm \frac{1}{2} \frac{108^{\frac{1}{6}}}{2} i.</math></p>

Таблица 6.2 (продолжение)

№ п/п	Уравнения	Ответы
7	$x^7 - ax^5 - x^2 - a = 0$	<p>a) <math>x_{1,2} = \pm\sqrt{-a}, x_3 = 1,</math>  <math>x_{1,2} = \pm\sqrt{-a}, x_3 = 1;</math></p> <p>b) <math>x_{4,5} = -\frac{\sqrt{5}}{4} - \frac{1}{4} \pm i\sqrt{\frac{5}{8} - \frac{\sqrt{5}}{8}},</math>  <math>x_{6,7} = \frac{\sqrt{5}}{4} - \frac{1}{4} \pm i\sqrt{\frac{5}{8} + \frac{\sqrt{5}}{8}}.</math></p>
8	$2^x - 2(a+b) = 0$	<p>a) <math>x = \frac{\ln(a+b)}{\ln 2} + 1;</math></p> <p>b) <math>x_1 = \frac{\ln(a+b)}{\ln 2} + 1,</math></p> <p>5. <math>x_{2,3} = \frac{\ln(a+b)}{\ln 2} + 1 \pm \frac{2\pi}{\ln 2}i.</math></p>
9	$\ln \sin x - 2a = 0$	$x_1 = a \sin e^{2a}, x_{2,3} = \pm\pi - a \sin e^{2a}$

Таблица 6.2 (продолжение)

№ п/п	Уравнения	Ответы
10	$e^{-2x} - 2a = 0$	а) $x = -\frac{\ln 2a}{2}$ ; б) $x_1 = -\frac{\ln 2a}{2}, x_{2,3} = -\frac{\ln 2a}{2} \pm \pi i$ .
11	$\sin ax + \cos ax = 0$	$x_1 = -\frac{5\pi}{4a}, x_2 = \frac{3\pi}{4a}, x_3 = -\frac{\pi}{4a}$
12	$\sin ax + \operatorname{tg} ax = 0$	$x_1 = -\frac{3\pi}{a}, x_{2,3} = \pm \frac{\pi}{a}$
13	$e^{-ax} + e^{ax} - 1 = 0$	а) false; б) $x_{1,2} = \pm \frac{7\pi}{3a}i, x_{3,4} = \pm \frac{5\pi}{3a}i, x_{5,6} = \pm \frac{\pi}{3a}i$ .
14	$e^{-ax} + e^{ax} = 0$	а) false; б) $x_{1,2} = \pm \frac{5\pi}{2a}i, x_{3,4} = \pm \frac{3\pi}{2a}i, x_{5,6} = \pm \frac{\pi}{2a}i$ .

Таблица 6.2 (окончание)

№ п/п	Уравнения	Ответы
15	$e^{-ax} + e^{ax} - \ln a = 0$	$x_{1,2} = \frac{\ln \left( \frac{\ln a}{2} \pm \sqrt{\frac{\ln^2 a - 4}{2}} \right)}{a}$
16	$ae^{-x} + be^x = 0$	<p>a) <math>x = \frac{\ln \left( -\frac{a}{b} \right)}{2}</math>;</p> <p>b) <math>x_1 = \frac{\ln \left( -\frac{a}{b} \right)}{2}, x_{2,3} = \frac{\ln \left( -\frac{a}{b} \right)}{2} \pm \pi i</math>.</p>
17	$a \sin x + b \cos x = 0$	$x_{1,2} = -a \tan \left( \frac{b}{a} \right) \pm \pi, x_3 = -a \tan \left( \frac{b}{a} \right)$

## 6.3. Численные методы решения алгебраических и трансцендентных уравнений

Численные методы решения алгебраических и трансцендентных уравнений с помощью системы Derive 5 имеют несколько особенностей.

- Трансцендентное уравнение может иметь большое число корней, которые с помощью одной команды программа не сумеет найти. Для их определения необходимо указать системе область изоляции каждого корня.
- Корни трансцендентного уравнения могут быть вещественными и комплексными. Для их определения необходимо использовать различные модификации функции `SOLVE`. Типичным примером является решение уравнения  $\operatorname{tg}(\cos x) = 0$  (рис. 6.6).

```
#1: TAN(COS(x))
#2: SOLVE(TAN(COS(x)), x, Real)
#3: 
$$x = -\frac{3 \cdot \pi}{2} \vee x = -\frac{\pi}{2} \vee x = \frac{\pi}{2}$$

#4: NSOLVE(TAN(COS(x)), x, Real)
#5: 
$$x = 7.853981256$$

#6: NSOLVE(TAN(COS(x)), x)
#7: 
$$x = 7.853981256$$

#8: SOLVE(TAN(COS(x)), x)
#9: 
$$x = 2 \cdot \pi - i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi \vee x = -\pi + i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi \vee x = \pi - i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi$$


$$\vee x = \pi + i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi \vee x = -i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi \vee x = i \cdot \operatorname{LN}(\sqrt{\pi^2 - 1}) + \pi$$


$$\vee x = -\frac{3 \cdot \pi}{2} \vee x = -\frac{\pi}{2} \vee x = \frac{\pi}{2}$$

```

Рис. 6.6. Решение уравнения  $\operatorname{tg}(\cos x) = 0$

- Функция `SOLVE(f(x), x, Real)` выдала три вещественных корня (строки — #1, #2, #3), функции `NSOLVE(f(x), x, Real)` и `NSOLVE(f(x), x, Complex)` выдали лишь один корень (стро-

ки — #4, #5, #6, #7). Функция `SOLVE (f(x), x, Complex)` выдала шесть комплексных корней и три вещественных (строки #8, #9). Однако все они вместе взятые не нашли множества других корней (уравнение  $\operatorname{tg}(\cos x) = 0$  имеет бесчисленное множество корней).

- Функция `SOLVE` во всех ее модификациях не найдет решения, если уравнение представляет собой многочлен пятой степени и выше. Исключением являются случаи, когда многочлен может быть разложен на множители. Например, уравнение  $x^n - a$  будет решено. Это решение показано на рис. 6.7.

```
#1:  x5 - a
#2:  SOLVE(x5 - a, x)
#3:  x = a1/5 · [ - $\frac{\sqrt{5}}{4}$  -  $\frac{1}{4}$  - i ·  $\sqrt{\left(\frac{5}{8} - \frac{\sqrt{5}}{8}\right)}$  ] ∨ x = a1/5 · [ - $\frac{\sqrt{5}}{4}$  -  $\frac{1}{4}$  + i ·  $\sqrt{\left(\frac{5}{8} - \frac{\sqrt{5}}{8}\right)}$  ] ∨
      x = a1/5 · [  $\frac{\sqrt{5}}{4}$  -  $\frac{1}{4}$  - i ·  $\sqrt{\left(\frac{\sqrt{5}}{8} + \frac{5}{8}\right)}$  ] ∨ x = a1/5 · [  $\frac{\sqrt{5}}{4}$  -  $\frac{1}{4}$  + i ·  $\sqrt{\left(\frac{\sqrt{5}}{8} + \frac{5}{8}\right)}$  ] ∨ x
      = a1/5
#4:  NSOLVE(x5 - a, x)
#5:  NSOLVE(x5 - a, x)
#6:  NSOLVE(x5 - a, x, Real)
#7:  NSOLVE(x5 - a, x, Real)
#8:  SOLVE(x5 - a, x, Real)
#9:  x = a1/5
```

**Рис. 6.7.** Решение уравнения  $x^n - a$

Функция `SOLVE (f(x), x, Complex)` выдала все корни в аналитическом виде. Функции `NSOLVE (f(x), x, Complex)` и `NSOLVE (f(x), x, Real)` уравнение не решили, а функция `SOLVE (f(x), x, Real)` выдала один вещественный корень.

В табл. 6.3 приведены алгебраические и трансцендентные уравнения. Указано число корней трансцендентных уравнений.



**Таблица 6.3.** Число корней алгебраических и трансцендентных уравнений

№ п/п	Уравнения		Число корней
	Алгебраические	Трансцендентные	
1	$x^4 - 4x^3 - 10x^2 + 3x - 4 = 0$	$2 \sin(\ln x ) = 0$	6
2	$x^5 - 5x^2 + 4,5 = 0$	$\operatorname{atg}(\operatorname{tg} x) = 0$	$\infty$
3	$x^5 - x + 0,2 = 0$	$4 \exp\left(-\frac{1}{ x }\right) - 2 = 0$	3
4	$x^3 - 0,2x^2 + 0,5x + 1,5 = 0$	$\sin(\sin(x)) = 0$	$\infty$
5	$x^{10} - 1 = 0$	$2^x - 4x = 0$	2
6	$x^8 + 2x - 1,5 = 0$	$x! + 2x - 2 = 0$	4
7	$x^4 - 2x^2 + 8x + 1 = 0$	$\ln x + (x + 1)^3 = 0$	1
8	$x^3 + 4x^2 - 5x - 2 = 0$	$\operatorname{arctg}(x - 2) + x = 0$	1
9	$6x^8 - 2x^2 + 3 = 0$	$x^2 + 4 \sin x - 2 = 0$	2
10	$3,5x^5 - 2,8x^3 + 7,5x - 2,5 = 0$	$x - \ln(7 - 4x) = 0$	1

Таблица 6.3 (окончание)

№ п/п	Уравнения		Число корней
	Алгебраические	Трансцендентные	
11	$x^5 + x^4 + x^3 + x^2 + x + 1 = 0$	$e^{-2x} + \frac{3}{x} - 1 = 0$	2
12	$1,5x^5 + 17x - 21 = 0$	$e^{-6x} + 3x^2 - 18 = 0$	2
13	$2x^4 - 3,5x^2 + 3 = 0$	$\ln(4 - 2x) + x^2 - 2 = 0$	3
14	$17x^9 - 15x^7 + 13x^5 - 11x^3 + 9x - 7 = 0$	$3^x - 9x + 1 = 0$	2
15	$x^4 + 2x^3 - 1 = 0$	$2^x + \ln 2x - 5,6 = 0$	1
16	$x^x - 21x^2 + 55 = 0$	$2^x + e^{-x} - 5x + 1 = 0$	2
17	$x^7 + 2x^6 + 3x^5 + 4x^4 + 5x^3 + 6x^2 + 7x - 8 = 0$	$3x - e^x + 1,5 = 0$	2
18	$8x^7 - 7x^6 + 6x^5 - 5x^4 + 4x^3 - 3x^2 + 2x - 1 = 0$	$\frac{e^x}{2} - (x-1)^2 = 0$	1
19	$x^{50} - 1 = 0$	$x + \lg x - 0,5 = 0$	1
20	$x^7 + 6x^6 + x^5 - 4x^4 + x^3 - 2x^2 + x - 1 = 0$	$2x! - e^{-x} + 5 = 0$	4

Число корней алгебраических уравнений очевидно — оно равно степени многочлена. Решение уравнений является наилучшим способом изучения универсального программного математического средства символьной математики Derive 5.

Существует большое число методов определения корней алгебраических и трансцендентных уравнений, например, дихотомии, хорд, касательных, итераций и др. Алгоритм любого из этих методов представляет совокупность условий выбора начального приближения, расчетных соотношений и признака окончания вычислительного процесса.

Методы и алгоритмы определения корней уравнений изучаются в ряде учебных дисциплин, например, таких как вычислительная математика, информатика и др. Изучаются свойства алгоритмов, их достоинства и недостатки, способы реализации и области применения. К сожалению, существующие универсальные математические программные средства, в том числе и Derive 5, не имеют функций, реализующих эти методы. Более того, пользователь не знает, каким методом решается уравнение в данной системе. Учащимся приходится самим составлять программы для компьютера на одном из универсальных языков программирования и, путем решения задач, анализировать методы и алгоритмы решения уравнений. В результате такого анализа получают следующие характеристики алгоритмов:

- ☐ массовость;
- ☐ результативность;
- ☐ сложность;
- ☐ время реализации и др.

Одновременно учащиеся приобретают навыки в программировании. В этом процессе важную роль могут сыграть универсальные математические программные средства — Mathematica, Maple, Derive, Mathcad, Matlab. Об этом будет говориться в следующих главах.

Сейчас рассмотрим алгоритмы численных методов определения корней уравнений.

### 6.3.1. Метод дихотомии (половинного деления)

Сущность метода состоит в следующем. Предположим, что областью изоляции корня уравнения  $f(x) = 0$  является  $[a, b]$ . Тогда за первое приближение к искомому корню  $x$  принимается —  $x_1 = \frac{a+b}{2}$ . Затем вычисляются значения функции  $f(x)$  в точках  $a$  и  $x_1$  (или  $b$  и  $x_1$ ). Если  $f(a) \cdot f(x_1) < 0$ , то новой областью изоляции корня является  $[a, x_1]$ , в противном случае  $[b, x_1]$ . Равносильным является условие  $f(b) \cdot f(x_1) < 0$ . Если это условие выполняется, то новой областью изоляции будет  $[b, x_1]$ , в противном случае  $[a, x_1]$ .

Вторым приближением к искомому корню является —  $x_2 = \frac{a+x_1}{2}$ , если  $f(a) \cdot f(x_1) < 0$  или  $x_2 = \frac{b+x_1}{2}$ , если  $f(b) \cdot f(x_1) < 0$ . Затем вычисляются значения функции  $b$  при  $x = x_2$ , с последующей проверкой условия  $f(x_1) \cdot f(x_2) < 0$ , и т. д.

Признаком окончания вычислительного процесса в этом методе является одно из следующих условий:

$f(x_n) \leq \varepsilon$  или  $|f(x_n) - f(x_{n-1})| \leq \varepsilon$ , где  $\varepsilon$  — допустимая погрешность вычисления корня.

Достоинством метода является простота алгоритма и высокая точность определения корня. Медленная сходимость итераций — основной недостаток метода.

### 6.3.2. Метод хорд

Алгоритмом метода хорд является совокупность следующих соотношений:

□ условие выбора начального приближения:

$$x_0 = \begin{cases} a, & \text{если } f(a)f''(a) < 0 \text{ или } f(b)f''(b) > 0, \\ b, & \text{если } f(a)f''(a) > 0 \text{ или } f(b)f''(b) < 0; \end{cases}$$

□ расчетные соотношения:

$$x_n \begin{cases} x_{n-1} - \frac{(b-x_{n-1})f(x_{n-1})}{f(b)-f(x_{n-1})}, & \text{если } x_0 = a \\ x_{n-1} - \frac{(a-x_{n-1})f(x_{n-1})}{f(a)-f(x_{n-1})}, & \text{если } x_0 = b \end{cases};$$

□ признак окончания вычислительного процесса:

$$|x_n - x_{n-1}| \leq \varepsilon \quad \text{или} \quad |f(x_n)| \leq \varepsilon.$$

В алгоритме метода хорд приняты следующие обозначения:

- $[a, b]$  — область изоляции корня;
- $f(a), f(b)$  — значения функций уравнения  $f(x) = 0$  в точках  $a$  и  $b$ ;
- $f''(a), f''(b)$  — значения вторых производных функции  $f(x)$  в точках  $a$  и  $b$ ;
- $x_n$  — приближение корня уравнения  $f(x) = 0$ ,  $n = 1, 2, \dots$ ;
- $\varepsilon$  — погрешность вычисления корня уравнения.

Как видно из описания алгоритма, он является итерационным. Для его реализации, в виде программы для ЭВМ, необходимо знать:

- область  $[a, b]$  изоляции корня;
- вторую производную  $f''(x)$ ;
- значение начального приближения (если не существует аналитического выражения второй производной функции  $f(x)$ ).

Получить эти данные можно с помощью универсальных программных средств символьной математики, существенно облегчив труд учащегося.

Метод хорд дает возможность получить решение с необходимой точностью и с меньшим числом итераций, по сравнению с методом половинного деления.

Недостатками данного метода являются:

- сложность (в связи с необходимостью вычисления второй производной);
- неудовлетворительный признак окончания вычислительного процесса.

### Примечание

Последний недостаток объясняется тем, что уточнение корня на каждой из итераций происходит по признаку  $|x_n - x_{n-1}| \leq \varepsilon$ , в то время как сам корень ( $\bar{x}$ ) при этом не находится в области  $[x_n, x_{n-1}]$ , так как приближение к корню идет только от начального приближения  $a$  или  $b$ . Может оказаться, что абсолютная разность  $|x_n - x_{n-1}|$  мала и удовлетворяет условию окончания вычислительного процесса, но при этом корень  $\bar{x}$  далеко расположен от  $x_n$ .

Если признаком окончания вычислительного процесса считать условие  $|f(x)| \leq \varepsilon$ , то при этом может оказаться, что значение функции  $f(x)$  мало, а абсцисса  $x_n$  далеко находится от корня  $\bar{x}$ .

Отсутствие хорошего признака окончания вычислительного процесса может привести к вычислению корня уравнения  $f(x) = 0$  с погрешностью, превышающей  $\varepsilon$ , хотя оба условия окончания вычислительного процесса выполнены.

### 6.3.3. Метод касательных

Идея метода состоит в следующем. Выбирается произвольно значение  $x$ , принадлежащее функции  $f(x)$  уравнения  $f(x) = 0$ . Проводится касательная к функции в этой точке до пересечения ее с осью абсцисс. Точка пересечения касательной с осью абсцисс (обозначим ее  $x_1$ ) принимается за первое приближение корня.

Вычисляется значение функции  $f(x_1)$  в точке  $x_1$  и вновь проводится касательная в точке с координатами  $x_1, f(x_1)$ . Точка  $x_2$  пересечения касательной с осью абсцисс принимается за второе приближение корня уравнения  $f(x) = 0$  и т. д. Признаком окончания вычислительного процесса, как и в методе хорд, является выполнение одного из условий  $|x_n - x_{n-1}| \leq \varepsilon$  или  $|f(x)| \leq \varepsilon$ .

Легко получить следующую рекуррентную формулу вычисления приближений:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}, \quad (6.1)$$

где  $f'(x_{n-1})$  — производная функции  $f(x)$  в точке  $x_{n-1}$ .

Начальное приближение  $x_0$ , как и в методе хорд, зависит от вида функции  $f(x)$  и области изоляции корня  $[a, b]$ . При этом оказывается, что оно будет противоположным значению  $x_0$  в методе хорд. Если в методе хорд  $x_0 = a$ , то в методе касательных  $x_0 = b$ , и наоборот.

Алгоритмом метода касательных является совокупность следующих соотношений:

□ условие выбора начального приближения:

$$x_0 = \begin{cases} a, \text{ если } f(a)f''(a) > 0 \text{ или } f(b)f''(b) < 0, \\ b, \text{ если } f(a)f''(a) < 0 \text{ или } f(b)f''(b) > 0; \end{cases}$$

□ расчетное соотношение:

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})};$$

□ признак окончания вычислительного процесса:

$$|x_n - x_{n-1}| \leq \varepsilon \text{ или } |f(x_n)| \leq \varepsilon.$$

Из алгоритма видно, что для его реализации в виде компьютерной программы необходимо знать:

- область изоляции корня;
- аналитические выражения первой и второй производных;
- начальное приближение, если не существует аналитического выражения второй производной.

Их определение возможно с помощью компьютерных технологий, реализуемых в универсальных программных средствах символьной математики.

Ограничения метода касательных: метод нельзя реализовать на практике, если функция  $f(x)$  уравнения  $f(x) = 0$  не имеет первой производной. Например, уравнение  $2x! - e^{-x} + 5 = 0$  не может быть решено, так как функция  $x!$  не имеет производной.

Метод касательных имеет те же недостатки, что и метод хорд. По сравнению с методом хорд он более трудоемкий, так как требует вычисления в точках  $x_0, x_1, x_2, \dots, x_n$  не только значений функции  $f(x)$ , но и ее производной. Его достоинство: во многих случаях дает высокую точность результата при малом числе итераций.

### 6.3.4. Комбинированный метод (метод хорд и касательных)

Существенным недостатком методов хорд и касательных является неудовлетворительный признак окончания вычислительного процесса. Условия  $|x_n - x_{n-1}| \leq \varepsilon$  или  $|f(x_n)| \leq \varepsilon$  не всегда обеспечивают необходимую точность определения корня уравнения  $f(x) = 0$ .

Комбинированный способ позволяет устранить этот недостаток. Из описания методов хорд и касательных следует, что если один из способов дает значение корня с недостатком, то другой обязательно с избытком. Эта особенность методов дает возможность выработать хороший признак окончания вычислений и обеспечить необходимую точность результата.



Обозначим  $x_n^{(x)}$  —  $n$ -е приближение корня, вычисленное по методу хорд,  $x_n^{(k)}$  — по методу касательных.

Тогда для оценки погрешности вычисления корня целесообразно воспользоваться условием  $|x_n^{(x)} - x_n^k| \leq \varepsilon$ , так как достоверно известно, что в диапазоне  $x_n^{(x)} - x_n^k$  обязательно находится искомый корень.

Алгоритмом вычисления корней комбинированным методом является совокупность следующих соотношений:

□ условия выбора начального приближения:

$$x_0 = \begin{cases} a, & \text{если } f(a) \cdot f''(a) > 0 \quad \text{или } f(b) \cdot f''(b) < 0, \\ b, & \text{если } f(a) \cdot f''(a) < 0 \quad \text{или } f(b) \cdot f''(b) > 0; \end{cases}$$

□ расчетные соотношения:

$$x_n^{(k)} = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})};$$

$$x_n^{(x)} = \begin{cases} x_{n-1} - \frac{(b - x_{n-1})f(x_{n-1})}{f(a) - f(x_{n-1})}, & \text{если } x_0 = a, \\ x_{n-1} - \frac{(a - x_{n-1})f(x_{n-1})}{f(a) - f(x_{n-1})}, & \text{если } x_0 = b; \end{cases}$$

□ признак окончания вычислительного процесса:

$$|x_n^{(k)} - x_n^{(x)}| \leq \varepsilon.$$

Недостатком метода хорд и касательных является большая его трудоемкость. Однако при высокой производительности компьютера этот недостаток значения не имеет.

Существенным преимуществом метода является его способность обеспечить высокую точность определения корня при конечном числе итераций.

### 6.3.5. Метод итераций

Исходное уравнение  $f(x) = 0$  преобразуется к виду  $x = \varphi(x)$ . Затем берется из области изоляции корня  $[a, b]$  произвольное  $x_0$ , которое и принимается за начальное приближение корня. Приближения  $x_1, x_2, x_3, \dots, x_n$  вычисляются по соотношениям  $x_1 = \varphi(x_0)$ ,  $x_2 = \varphi(x_1)$ , ...,  $x_n = \varphi(x_{n-1})$ .

Повторяя эти процедуры многократно, можно вычислить значение корня с заданной точностью.

При практическом использовании метода итераций возникают следующие вопросы.

- Каковы условия сходимости итерационного процесса?
- Если итерационный процесс расходится, то каким образом можно обеспечить его сходимость?
- Как определить погрешность вычисления корня?

Существуют две теоремы, которые отвечают на первый вопрос. Приведем их без доказательства.

*Теорема 1.* Если в итерационном процессе  $x_n = \varphi(x_{n-1})$  последовательность  $x_1, x_2, x_3, \dots, x_n$  имеет предел, то есть  $\lim_{n \rightarrow \infty} x_n = \bar{x}$ , то значение  $\bar{x}$  является корнем уравнения  $f(x) = 0$ .

*Теорема 2.* Итерационный процесс сходится, если на всем интервале области изоляции корня  $[a, b]$  выполняется условие  $|\varphi'(x)| < 1$ . При этом за значение  $x_0$  принимается любое число из области  $[a, b]$ .

Теперь ответим на второй вопрос. Известны несколько способов обеспечения сходимости итераций.

*Способ 1.* Если итерационный процесс  $x_n = \varphi(x_{n-1})$  не сходится, то следует представить исходное уравнение  $f(x) = 0$  в такой возможной форме, при которой обеспечивается сходимость итерационного процесса.

*Способ 2.* Переход к обратной функции.

Представим исходное уравнение  $x = \varphi(x)$  в виде  $y = \varphi(x)$  и решим его относительно  $x$ . Получим  $x = \Psi(y)$ . Теперь найдем производную по  $y$  функции  $x = \Psi(y)$ :

$$\frac{d\Psi(y)}{dy} = \frac{dx}{dy} = \frac{1}{\frac{dy}{dx}} = \frac{1}{\varphi'(x)}.$$

Так как при расходящемся итерационном процессе  $|\varphi'(x)| > 1$ , то  $\left| \frac{1}{\varphi'(x)} \right| < 1$  и итерационный процесс  $y_n = f(y_{n-1})$  сходится. Очевидно, что если  $x_k$  — корень уравнения  $y = \Psi(y)$ , то он также будет корнем уравнения  $x = \varphi(x)$ .

*Способ 3.* Подбор множителя.

Предположим, что исходное уравнение  $f(x) = 0$  преобразовано к виду  $x = \varphi(x)$  и  $|\varphi'(x)| > 1$ , то есть процесс расходится. Выберем произвольно функцию  $g(x) \neq 0$  и умножим исходное уравнение на  $g(x)$ . Тогда получим —  $f(x) \cdot g(x) = 0$  или  $x = x - f(x) \cdot g(x)$ , то есть  $\varphi(x) = x - f(x) \cdot g(x)$ . Подберем функцию  $g(x)$  такую, чтобы удовлетворялось условие  $|\varphi'(x)| < 1$  во всей области изоляции корня  $[a, b]$ .

Признаком окончания вычислительного процесса во всех предыдущих методах было одно из условий —  $|x_n - x_{n-1}| \leq \varepsilon$  или  $|f(x_n)| \leq \varepsilon$ .

В методе итераций условием сходимости итерационного процесса и обеспечения необходимой точности определения корня является:

$$|x_n - x_{n-1}| \leq \varepsilon \text{ при } |\varphi'(x)| < \frac{1}{2}.$$

Из описания метода итераций можно сформулировать следующий алгоритм решения уравнения  $f(x) = 0$  :

- условие выбора начального приближения:

$$a \leq x_0 \leq b;$$

- расчетное соотношение:

$$x_n = \varphi(x_{n-1}) \text{ при условии } |\varphi'(x)| < \frac{1}{2};$$

- признак окончания вычислений

$$|x_n - x_{n-1}| \leq \varepsilon.$$

Из описания метода итераций видно, что основным его недостатком является сложность обеспечения сходимости итерационного процесса и точности определения корня.

## 6.4. Компьютерные технологии решения уравнений

Из описания методов дихотомии, хорд, касательных и итераций следует, что разработка программ определения корней уравнений требует в ряде случаев от программиста знания:

- области изоляции корня;
- первой и второй производных функций  $f(x)$ ;
- значения начального приближения;
- проверки условия и обеспечения сходимости итерационного процесса.

Только при этих условиях могут быть составлены программы перечисленных методов и проведены исследования алгоритмов. Определить область изоляции корня и вычислить значения производных и, тем более, проверить и обеспечить сходимость итерационного процесса "вручную" чрезвычайно трудно.

Время, затраченное на эти процедуры, будет превосходить время, необходимое для составления программы решения уравнения.

Здесь следует использовать универсальные программные средства символьной математики, в частности систему Derive 5. Ниже описаны технологии решения ряда задач при реализации методов определения корней уравнений.

### 6.4.1. Определение области изоляции корня

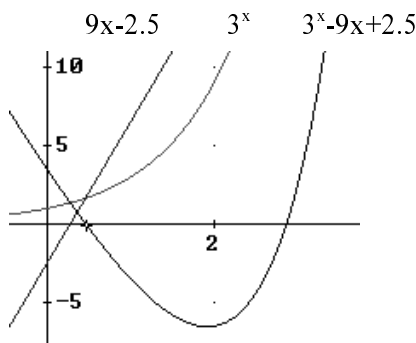
*Способ 1.* Пусть функция  $f(x)$ , в области изоляции корня  $[a, b]$ , непрерывна. Тогда условием существования хотя бы одного корня является отношение —  $f(a) \cdot f(b) < 0$ , т. е. значения функций  $f(a)$  и  $f(b)$  должны быть с разными знаками. Реализация этого метода в системе Derive 5 чрезвычайно проста и очевидна.

Необходимо ввести функцию  $f(x)$  и с помощью команды **Substitute for Variables** (кнопка **Sub** на панели инструментов) присвоить переменным  $a$  и  $b$  численные значения. После команды **Approximate** — на экране значения функций  $f(a)$  и  $f(b)$ . Если они разных знаков, то в области  $[a, b]$  имеется хотя бы один корень. Этот метод имеет следующие два недостатка: область изоляции выбирается методом проб и ошибок, а в полученной области  $[a, b]$  может быть несколько корней.

*Способ 2.* Функция  $f(x)$  представляется графически, и по виду графика визуально определяется число корней и их области изоляции. Иногда бывает полезным этот способ модифицировать, для чего представить функцию  $f(x)$  в виде  $f(x) = f_1(x) + f_2(x)$ . Далее сформировать на одном графике функции  $f_1(x)$  и  $f_2(x)$ . Абсциссы точек пересечения их будут корнями уравнения  $f(x) = 0$ .

Реализация данного способа в Derive 5 довольно проста и способ исключительно наглядный. Вводится функция  $f(x)$  и с помощью кнопок панели инструментов **2D-plot window | Plot Expression** на экране формируется график функций. Абсциссы точек пересечения с осью  $x$  и являются корнями уравнения  $f(x) = 0$ . Далее по виду графика определяются области изоляции всех корней.

Описанные способы показаны на рис. 6.8 для случаев решения трансцендентного уравнения  $3^x - 9x + 2,5 = 0$ .



**Рис. 6.8.** Определение области изоляции корня

Из рис. 6.8 следует, что уравнение имеет два корня, областями изоляции которых могут быть  $[0, 1]$  и  $[2, 4]$ . На рисунке также показаны графики функций  $y = 3^x$  и  $y = 9x - 2,5$ . Абсцисса точки пересечения этих функций совпадает с абсциссой точки пересечения функции  $3^x - 9x + 2,5$  с осью  $x$ . В нашем примере корнями уравнения являются  $x_1 = 0,4624\dots$ ,  $x_2 = 2,865\dots$ .

При выборе области изоляции следует помнить, что чем она уже, тем меньше потребуются итераций для определения корня.

## 6.4.2. Определение производных

Рассмотренные методы требуют отыскания первой и второй производных в аналитическом виде. Технологии определения производных в системе Derive 5 подробно описаны в гл. 4. Следует только иметь в виду, что не всякая функция имеет производные. Например, функции, в состав которых входят факториалы неизвестных — производных не имеют. Уравнение  $2x! - 3,5x + 2 = 0$  имеет два вещественных корня  $x_1 = 0,098\dots$  и  $x_2 = 5,069\dots$ . Однако это уравнение методом касательных, комбинированным методом

и методом итераций не может быть решено, так как все эти методы требуют вычисления производных. Методом хорд решить это уравнение можно только в том случае, если начальное приближение  $x_0$  будет определено программистом и его не нужно искать программе, проверяя одно из условий —  $f(a) \cdot f''(a) < 0$  или  $f(b) \cdot f''(b) < 0$ .

Метод половинного деления не требует знания начального приближения и производных функции  $f(x)$ , поэтому его можно применить для решения приведенного уравнения. Следует только иметь в виду, что вычисление значения  $x!$  потребует многократного вычисления значения гамма-функции.

### 6.4.3. Обеспечение сходимости итераций

Ранее были описаны три способа обеспечения сходимости итераций, при решении уравнения методом итераций. Здесь Derive 5 может оказать существенную помощь программисту в подборе множителя, в реализации способа перехода к обратной функции и в приведении уравнения  $f(x) = 0$  к виду  $x = \varphi(x)$ .

### Реализация метода итераций в системе Derive 5

Итерационные алгоритмы в системе Derive 5 реализуются функцией `ITERATES`. Эта функция имеет вид:

`ITERATES (f(x), x, x0, n),`

где  $f(x)$  — функция, определяемая исходным уравнением  $x = f(x)$ ;  $x$  — искомое неизвестное;  $x_0$  — начальное приближение;  $n$  — число итераций, задаваемое пользователем. Функция `ITERATES` выдает решение в виде результатов во всех  $n$  итерациях. Она также может иметь вид:

`ITERATE (f(x), x, x0, n).`

В этом случае возвращается конечный результат, т.е. результат конечной итерации.

Из описания функции видно, что для решения уравнения необходимо выполнить следующие действия:

- определить число корней, их области изоляции и начальные приближения;
- разрешить исходное уравнение, представив его в следующем виде —  $x = f(x)$ ;
- определить необходимое число итераций  $n$ .

Разрешение уравнения относительно неизвестного возможно многими способами. Пусть, например, нелинейное уравнение имеет вид —  $3^x + 2x - 1,2 = 0$ . Представить это уравнение в виде

$$x = f(x) \quad \text{можно следующими способами:} \quad x = \frac{1,2 - 3^x}{2},$$

$$x = \frac{\ln(1,2 - 2x)}{\ln(3)}.$$

Какое же из этих уравнений следует использовать для получения решения методом итераций? Пригодной для реализации метода итераций будет та функция  $f(x)$ , которая удовлетворяет условию —  $|f'(x)| < 1$  во всей области изоляции

$$\text{корня. В данном случае } f_1(x) = \frac{1,2 - 3^x}{2}, \quad f_2(x) = \frac{\ln(1,2 - 2x)}{\ln(3)}.$$

Производные этих функций, найденные с помощью пункта меню **Calculus | Differentiate** или команды **Find Derivative** ( $\partial$ ), на панели инструментов имеют вид:

$$\left| f_1'(x) \right| = \frac{3^x \ln(3)}{2}, \quad \left| f_2'(x) \right| = \frac{5}{(5x - 3) \ln(3)}.$$

Областью изоляции является  $x = [0 - 1]$ . В этом диапазоне производные имеют значения:

$$\square \quad |f_1'(x)| = 0,549 - 0,61;$$

$$\square \quad |f_2'(x)| = 1151 - 1,82.$$



Так как только функция  $f_1(x)$  удовлетворяет условиям сходимости итераций  $|f'(x)| < 1$  во всей области изоляции корня, то исходное уравнение следует представить в виде:

$$x = \frac{1,2 - 3^x}{2}.$$

#### 6.4.4. Определение числа итераций

Признаком окончания итераций является условие —  $|x_n - x_{n-1}| \leq \varepsilon$ . Функция `ITERATES` не дает возможности автоматически установить число итераций из этого условия. Для обеспечения необходимой точности вычисления корня пользователю самому приходится сравнивать значения корня при различном числе итераций и принимать решение о конечном значении корня. При таком методе целесообразно проверять правильность решения методом его подстановки в исходное уравнение. Рассмотрим описанную технологию на примере определения корней трансцендентного уравнения.

*Пример 6.5.* Пусть необходимо определить корни уравнения  $2^x - 4x = 0$  с точностью четыре знака после запятой.

#### Определение числа корней и областей их изоляции

Воспользуемся графическим способом. После набора и ввода уравнения на экране будет воспроизведено уравнение в следующем виде:

$$\#1: 2^x - 4x$$

Щелчок мыши по кнопке **2D-Plot Window**, расположенной на панели инструментов, приведет к формированию окна с пустой сеткой координат, графическим меню и собственной панелью инструментов. Щелчок мышью по кнопке **Plot Expression** сформирует на экране график функции (рис. 6.9).

Из графика следует, что уравнение имеет два корня (точки пересечения с осью абсцисс). Их областями изоляции могут быть:  $[0—1]$  и  $[3—5]$ . За начальные приближения возьмем  $x_0 = 0,5$  и  $x_0 = 3$ .

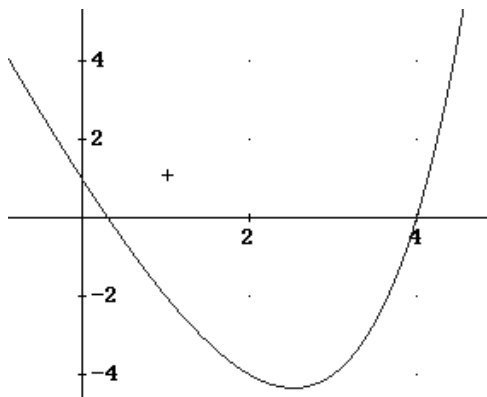


Рис. 6.9. График функции  $y = 2^x - 4x$

## Представление уравнения в виде $x = f(x)$

Уравнение  $2^x - 4x = 0$  может быть представлено в виде  $x = f(x)$  следующими двумя способами:

□  $x = 2^{x-2};$

□  $x = 2 + \frac{\ln(x)}{\ln(2)}.$

Проверим условие сходимости итераций в области изоляции корня  $[0—1]$  и  $[3—5]$ . Для этого вычислим производные функций

$f_1(x) = 2^{x-2}$  и  $f_2(x) = 2 + \frac{\ln(x)}{\ln(2)}$ . Наберем и введем эти функции.

На экране они будут представлены в следующем виде:

#2:  $2^{x-2}$

#3:  $2 + \frac{\ln(x)}{\ln(2)}$

Определим производные этих функций, воспользовавшись пунктом меню **Calculus | Differentiate** или кнопкой **Find Derivative** ( $\partial$ ), расположенной на панели инструментов. В появившемся

окне **Calculus Differentiate** будет определена переменная  $x$  (панель **Variable**) и первая производная (панель **Order**). Согласимся с программой и щелкнем мышью по кнопке **Simplify**. В результате на экране появятся следующие функции:

$$\#4: \frac{d}{dx} 2^{x-2}$$

$$\#5: 2^{x-2} \text{LN}(2)$$

$$\#6: \frac{d}{dx} \left( 2 + \frac{\text{LN}(x)}{\text{LN}(2)} \right)$$

$$\#7: \frac{1}{x \text{LN}(2)}$$

Функция  $|f_1'(x)| = |2^{x-2} \text{LN}(2)|$  при  $x=3$  и  $x=5$  имеет значения: 1,386 и 5,545, что больше единицы. При  $x=0$  и  $x=1$  производная соответственно равна — 0,173 и 0,346, что меньше единицы.

Функция  $|f_2'(x)| = \left| \frac{1}{x \text{LN}(2)} \right|$  при  $x=3$  и  $x=5$  имеет значения: 0,48 и 0,288, что меньше единицы. При  $x=0$  и  $x=1$  производная соответственно равна —  $\pm\infty$  и 1,44, что больше единицы.

Эти вычисления выполнены с помощью пункта меню **Simplify** и команды **Variable Substitution** путем подстановки значений  $x$  в соответствующие выражения. Можно также воспользоваться кнопкой **Sub** на панели инструментов.

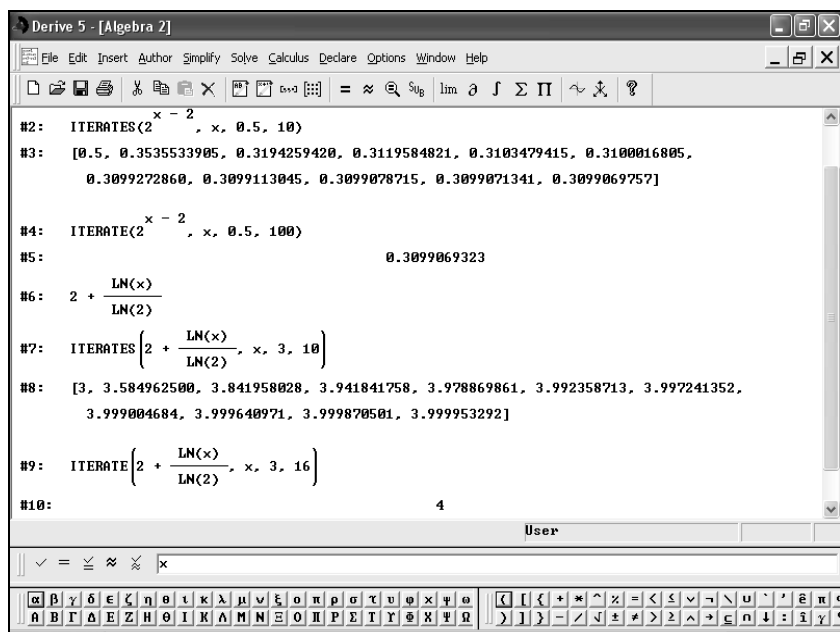
Из расчетов следует, что для определения методом итераций значения корня из области изоляции  $[0—1]$  исходное уравнение необходимо представить в виде:  $x = 2^{x-2}$ . Для определения корня из области изоляции  $[3—5]$  уравнение должно быть представлено так:  $x = 2 + \frac{\ln(x)}{\ln(2)}$ . Решим теперь уравнение методом итераций и определим их число. Для этого наберем и введем функцию:

`ITERATES (2^(x-2), x, 0.5, 10).`

Щелкнем по кнопке **Approximate** на панели инструментов. В результате на экране появится ответ в виде вектора значений десяти итераций. Из ответа видно, что значение корня с точностью четыре значащих цифры, после запятой, получено уже на седьмой итерации. Увеличим число итераций до  $n=100$  и вычислим значение корня с помощью функции:

`ITERATE(2^(x-2), x, 0.5, 100).`

В этом случае значение корня практически совпадает со значением, полученным при числе итераций  $n=10$ . Решение показано на рис. 6.10.



**Рис. 6.10.** Решение уравнения  $2^x - 4x = 0$  методом итераций

Вычислим теперь корень из области  $[3-5]$ . Для этого наберем и введем функцию:

`ITERATES(2+LN(x)/LN(2), x, 3, 10).`

Щелкнем по кнопке **Approximate** ( $\approx$ ). В результате на экране будет сформирован ответ в виде вектора значений десяти ите-

раций. На десятой итерации будет получен ответ:  $x=3.99995$ . Точное значение корня —  $x=4$ . Это значение будет получено при числе итераций  $n=16$  и числе знаков ответа равном 10. Решение уравнения  $2^x - 4x = 0$  показано на рис. 6.10.

Методы хорд и касательных являются также итерационными методами. Поэтому их реализация возможна с помощью функции `ITERATES`. В этом случае в функции `ITERATES (f, x, x0, n)` переменные будут иметь значения:

$$f = \begin{cases} x - \frac{(b-x)f(x)}{f(b)-f(x)}, & \text{если } x_0 = a \\ x - \frac{(a-x)f(x)}{f(a)-f(x)}, & \text{если } x_0 = b \end{cases}$$

Для определения начального приближения  $a$  или  $b$  необходимо вычислить произведение  $f(a)f''(a)$ . Если это произведение меньше нуля, то начальное приближение  $x_0 = a$ , в противном случае  $x_0 = b$ .

Вычисленное значение  $x_0$  подставляется в соответствующее выражение функции `ITERATES`.

В качестве примера вычислим методом хорд корни уравнения  $2^x - 4x = 0$ . Последовательность решения задачи в системе `Derive 5` имеет следующий вид.

□ Набор и ввод выражения  $2^x - 4x$  — на экране выражение:

#1:  $2x-4x$

□ Вычисление второй производной выражения #1:. Для этого: щелкнуть мышью по кнопке  $\partial$  на панели инструментов — на экране появляется окно **Calculus Differentiate**, в которое вводится переменная  $x$  и на вкладке **Order** цифра 2 (вторая производная). После щелчка мышью по кнопке **Simplify** на экране выражения:

#2:  $\left(\frac{d}{dx}\right)^2(2^x - 4x)$

#3:  $2^x \ln(2)^2$

- Набор и ввод выражения  $\#1*\#3$  — на экране появляется следующее выражение:

$$\#4: (2^x - 4x) 2^x \ln(2)^2$$

- Определение начальных приближений. Подставляя в выражение  $\#4$ , с помощью команды **Sub**, последовательно значения  $x = 0$  и  $x = 3$ , получим:

$$\#5: \ln(2)^2$$

$$\#6: -32 \ln(2)^2$$

Из выражений  $\#5$  и  $\#6$  видно, что начальным приближением при вычислении корня уравнения из области  $[0—1]$  будет  $x_0 = b = 1$ , так как  $\ln(2)^2 > 0$ . Начальным приближением при вычислении корня уравнения из области  $[3—5]$  будет  $x_0 = a = 3$ , так как при  $x = 3 - 32 \ln(2)^2 < 1$ . При полученных значениях начальных приближений расчетные соотношения будут иметь вид:

$$f_1 = x - \frac{(0-x)f(x)}{f(0)-f(x)} = x + \frac{x(2^x - 4x)}{1 - 2^x + 4x} \text{ при вычислении корня из области } [0—1],$$

$$f_2 = x - \frac{(5-x)(2^x - 4x)}{2^5 - 4 \cdot 5 - 2^x + 4x} = x - \frac{(5-x)(2^x - 4x)}{12 - 2^x + 4x} \text{ при вычислении корня из области } [3—5].$$

Вычисление корня, по выражению для  $f_1$ , будем осуществлять при начальном значении  $x_0 = 1$ , а по выражению  $f_2$  — при  $x_0 = 3$ . Функция **ITERATES** при этом будет иметь вид:

- **ITERATES**( $f_1, x, 1, n$ ) — при вычислении корня из области  $[0—1]$ ;

- **ITERATES**( $f_2, x, 3, n$ ) — при вычислении корня из области  $[3—5]$ .

Решение уравнения при определении корня из области изоляции  $[0—1]$  показано на рис. 6.11 и 6.12.

```

#1:   $2^x - 4 \cdot x$ 
#2:  NSOLVE( $2^x - 4 \cdot x$ ,  $x$ , 0, 1)
#3:                                      $x = 0.3099069323$ 
#4:   $x + \frac{x \cdot (2^x - 4 \cdot x)}{1 - 2^x + 4 \cdot x}$ 
#5:  ITERATES  $\left( x + \frac{x \cdot (2^x - 4 \cdot x)}{1 - 2^x + 4 \cdot x}, x, 1, 10 \right)$ 
#6:  [1, 0.3333333333, 0.3105361644, 0.3099237098, 0.3099073796, 0.3099069443, 0.3099069326,
      0.3099069323, 0.3099069323, 0.3099069323, 0.3099069323]

```

**Рис. 6.11.** Решение уравнения при  $n = 10$ 

```

#7:  ITERATE  $\left( x + \frac{x \cdot (2^x - 4 \cdot x)}{1 - 2^x + 4 \cdot x}, x, 1, 100 \right)$ 
#8:                                      $0.3099069323$ 

```

**Рис. 6.12.** Решение уравнения при  $n = 100$ 

В строках #1, #2, #3 приведены результаты определения корня с помощью функции NSOLVE. В строках #4, #5, #6 приведено решение уравнения методом хорд при числе итераций  $n=10$ . В строках #7, #8 определен корень уравнения при фиксированном числе итераций  $n=100$ . Из сравнения результатов решения уравнения видно, что метод хорд выдал решение с точностью 10 знаков после запятой уже на седьмой итерации.

Процедуры определения корня уравнения из области изоляции [3—5] показаны на рис. 6.13.

```

#1:   $x - \frac{(5 - x) \cdot (2^x - 4 \cdot x)}{12 - 2^x + 4 \cdot x}$ 
#2:  ITERATES  $\left( x - \frac{(5 - x) \cdot (2^x - 4 \cdot x)}{12 - 2^x + 4 \cdot x}, x, 3, 20 \right)$ 
#3:  [3, 3.5, 3.774367239, 3.903559981, 3.959802512, 3.983426085, 3.993197368, 3.997213152,
      3.998859186, 3.999533148, 3.999808976, 3.999921842, 3.999968022, 3.999986916,
      3.999994647, 3.999997809, 3.999999103, 3.999999633, 4, 4, 4]

```

**Рис. 6.13.** Процедуры определения корня уравнения из области изоляции [3—5]

Решение получено при  $x_0 = 3$  и числе итераций  $n = 20$ . Из решения видно, что уже на восемнадцатой итерации получено точное значение корня:  $x = 4$ .

## Реализация в Derive 5 метода касательных

Переменная  $f$  в функции `ITERATES`, в случае реализации метода касательных, в соответствии с (6.1) имеет вид:  $f = x - \frac{f(x)}{f'(x)}$ . Начальное приближение определяется аналогично методу хорд.

Технологию решения уравнения рассмотрим на предыдущем примере. Итак, необходимо решить уравнение  $2^x - 4x = 0$  и сравнить с результатами решения по методу хорд и итераций.

Составим программу решения задачи:

- ввод выражения  $2^x - 4x$ , на экране выражение: #1:  $2^x - 4x$ ;
- определение производной с помощью команды **Find Derivative** ( $\partial$ ) — на экране формируется производная в виде следующих выражений:

$$\#2: \left(\frac{d}{dx}\right)^2(2^x - 4x)$$

$$\#3: 2^x \ln(2) - 4;$$

- ввод выражения  $x - \#1/\#3$  — на экране формируется выражение:

$$\#4: x - \frac{2^x - 4x}{2^x \ln(2) - 4};$$

- ввод функции `ITERATES` ( $\#4, x, r, n$ ) — на экране формируется выражение:

$$\#5: \text{ITERATES}\left(x - \frac{2^x - 4x}{2^x \ln(2) - 4}, x, r, n\right).$$

Получена функция, позволяющая найти корни уравнения. Для этого необходимо с помощью команды **Sub** ввести начальное



значение  $r$  и желаемое число итераций  $n$ . В нашем случае начальным приближением будет  $r = 0$ , при вычислении корня из области  $[0—1]$ , и  $r = 5$ , при вычислении корня из области  $[3—5]$ . После ввода значений  $r$  и  $n$  достаточно щелкнуть мышью по кнопке **Approximate** ( $\approx$ ), расположенной на панели инструментов, и получить решение. Решение задачи показано на рис. 6.14 и 6.15.

$$\begin{aligned}
 \#1: & 2^x - 4 \cdot x \\
 \#2: & \frac{d}{dx} (2^x - 4 \cdot x) \\
 \#3: & 2^x \cdot \ln(2) - 4 \\
 \#4: & x - \frac{2^x - 4 \cdot x}{2^x \cdot \ln(2) - 4} \\
 \#5: & \text{ITERATES} \left( x - \frac{2^x - 4 \cdot x}{2^x \cdot \ln(2) - 4}, x, r, n \right) \\
 \#6: & \text{ITERATES} \left( x - \frac{2^x - 4 \cdot x}{2^x \cdot \ln(2) - 4}, x, 0, 10 \right)
 \end{aligned}$$

**Рис. 6.14.** Решение уравнения  $2^x - 4x = 0$  при  $r = 0$

$$\begin{aligned}
 \#7: & [0, 0.3024023307, 0.3099016185, 0.3099069323, 0.3099069323, 0.3099069323, 0.3099069323, \\
 & 0.3099069323, 0.3099069323, 0.3099069323, 0.3099069323] \\
 \#8: & \text{ITERATES} \left( x - \frac{2^x - 4 \cdot x}{2^x \cdot \ln(2) - 4}, x, 3, 10 \right) \\
 \#9: & [3, 5.588699449, 4.710905999, 4.191888106, 4.017849970, 4.000170826, 4, 4, 4, 4, 4]
 \end{aligned}$$

**Рис. 6.15.** Решение уравнения  $2^x - 4x = 0$  при  $r = 5$

Из решения видно, что ответ, с точностью 10 знаков после запятой, получен уже на третьей итерации, при определении корня из области  $[0—1]$ , и на седьмой, при определении корня из области  $[3—5]$ . Таким образом, метод касательных, при решении уравнения  $2^x - 4x = 0$ , оказался наиболее экономичным по сравнению с методом хорд и итераций, а при одинаковом числе итераций наиболее точным.

## Глава 7



# Решение систем алгебраических уравнений

Системы уравнений могут быть линейные и нелинейные, с постоянными и переменными коэффициентами. Решение таких уравнений возможно аналитическими и численными методами. Аналитические методы дают решение в общем виде. Такое решение, в силу своей общности, всегда более предпочтительно, чем численное, которое дает ответ лишь для конкретных числовых коэффициентов.

Однако решение уравнений аналитическими методами крайне ограничено. Они могут быть применены лишь для решения систем линейных алгебраических уравнений и, в редких случаях, для решения систем нелинейных уравнений невысокого порядка.

Система Derive 5 имеет богатые возможности решения систем уравнений аналитическими и численными методами.

Рассмотрим сначала математические методы решения линейных уравнений. Это поможет более квалифицированно и глубоко понять компьютерные технологии решения уравнений в среде Derive 5.

## 7.1. Решение систем линейных алгебраических уравнений

Систему  $m$  линейных уравнений с  $n$  неизвестными можно представить в следующем виде:

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ ..... \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{array} \right\} \quad (7.1)$$

В этой системе  $x_1, x_2, \dots, x_n$  неизвестные,  $a_{ij}$  — коэффициент в уравнении  $i$  при неизвестном  $j$  ( $i = 1, 2, \dots, m$ ;  $j = 1, 2, \dots, n$ ),  $b_i$  — свободный член уравнения  $i$ ,  $i = 1, 2, \dots, m$ .

В системах уравнений возможны случаи, когда число уравнений равно числу неизвестных ( $m = n$   $m = n$ ), меньше числа неизвестных ( $m < n$   $m < n$ ), больше числа неизвестных ( $m > n$   $m > n$ ). При этом, решением системы (7.1) является любой набор чисел  $x_1, x_2, \dots, x_n$ , которые, при подстановке в эту систему, обращают каждое из уравнений в числовое равенство.

Количество решений может быть равно, может быть меньше или больше числа неизвестных.

В зависимости от этого система уравнений классифицируется следующим образом. Система называется совместной, если она имеет хотя бы одно решение, в противном случае она называется несовместной. Совместная система может иметь единственное решение или бесконечное их число. Если имеется бесконечное число решений, то систему называют неопределенной. Рассмотрим примеры.

*Пример 7.1.*

Система уравнений:

$$\begin{aligned}x_1 + 2x_2 - x_3 &= 1; \\ 2x_1 + x_2 + x_3 &= 2; \\ 3x_1 + 2x_2 + 3x_3 &= 5;\end{aligned}$$

имеет единственное решение —  $x_1 = 0, x_2 = 1, x_3 = 1$ . Это значит, что она является совместной и определенной.

Система уравнений:

$$x_1 + 2x_2 - x_3 = 1;$$

$$2x_1 + 4x_2 - 2x_3 = 2;$$

$$3x_1 + 2x_2 + 3x_3 = 5;$$

имеет бесконечное число решений, которые удовлетворяют следующим равенствам:  $x_1 + 2x_3 = 2, 2x_2 - 3x_3 = -1$ . Другими словами, система является совместной и неопределенной. В ней определены лишь условия решения, причем первое и второе уравнения системы — идентичны.

Система:

$$x_1 + 2x_2 - x_3 = 1;$$

$$2x_1 + 4x_2 - 2x_3 = 2;$$

$$3x_1 + 6x_2 - 3x_3 = 3;$$

не имеет ни одного решения и является несовместной. Derive 5 выдает решение в виде  $x_1 + 2x_2 - x_3 = 1$ . Действительно, в этой системе будет три одинаковых уравнения (после деления второго уравнения на 2, а третьего на 3).

Существует большое число различных методов решения систем линейных уравнений. Все они могут быть разделены на две группы:

□ Точные методы.

□ Методы последовательных приближений.

Следует при этом иметь в виду, что точными методами являются только аналитические. Если с помощью этих методов решать систему уравнений с числовыми коэффициентами, то точных решений можно и не получить, за счет ошибок вычислений, связанных с ограниченным объемом памяти компьютера.

Наиболее популярным из точных методов решения линейных алгебраических уравнений является метод Гаусса. Данный метод

изучается в математике, и нет надобности его здесь описывать. Напомним только теорему Крамера.

Если определитель  $|A|$  матрицы коэффициентов системы  $n$  линейных уравнений с  $n$  неизвестными отличен от нуля, то система имеет решение и притом единственное.

При решении системы  $n$  линейных уравнений необходимо выполнить следующее количество операций:

$$N = \frac{2n(n+1)(n+2)}{3} + n(n-1).$$

Если, например,  $n = 10$ , то число операций будет  $N = 970$ .

Метод Гаусса может привести к существенным ошибкам при определении неизвестных  $x_1, x_2, \dots, x_n$  в случае плохо обусловленных систем. Плохо обусловленной называется такая система, у которой модуль определителя матрицы коэффициентов мал по сравнению с какой-либо из норм матрицы. Нормой матрицы может быть максимальная из сумм модулей коэффициентов строк или столбцов. Плохо обусловленные системы чувствительны к ошибкам округления, которые неизбежны при компьютерных методах реализации алгоритма Гаусса.

Рассмотрим решение линейных алгебраических уравнений методом простой итерации.

Разрешим исходную систему уравнений (7.1) относительно неизвестных:

$$\begin{aligned} x_1 &= \alpha_{1,2}x_2 + \alpha_{1,3}x_3 + \dots + \alpha_{1,n}x_n + \alpha_{1,n+1}; \\ x_2 &= \alpha_{2,1}x_1 + \alpha_{2,3}x_3 + \dots + \alpha_{2,n}x_n + \alpha_{2,n+1}; \\ &\dots \\ x_n &= \alpha_{n,1}x_1 + \alpha_{n,2}x_2 + \dots + \alpha_{n,n-1}x_{n-1} + \alpha_{n,n+1}. \end{aligned} \tag{7.2}$$

В системе приняты обозначения:  $\alpha_{i,k} = \frac{\alpha_{i,k}}{\alpha_{i,i}}$ , где  $i = 1, 2, \dots, n$ ;  $k = 1, 2, \dots, n, n+1$ .

Запишем систему уравнений (7.2) в следующем виде:

$$\begin{aligned}x_1 &= \varphi_1(x_1, x_2, \dots, x_n); \\x_2 &= \varphi_2(x_1, x_2, \dots, x_n); \\&\dots \\x_n &= \varphi_n(x_1, x_2, \dots, x_n).\end{aligned}\tag{7.3}$$

Пусть  $x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}$  являются начальными приближениями.

Тогда, подставляя их в систему уравнений (7.3), получим:

$$\begin{aligned}x_1^{(1)} &= \varphi_1(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}); \\x_2^{(1)} &= \varphi_2(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}); \\&\dots \\x_n^{(1)} &= \varphi_n(x_1^{(0)}, x_2^{(0)}, \dots, x_n^{(0)}).\end{aligned}\tag{7.4}$$

Принимая  $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$  в качестве первых приближений и подставляя их в исходное уравнение, получим вторые приближения. Повторяя вычисления, можно получить значения неизвестных на любой итерации.

При компьютерной реализации итерационного метода возникают следующие вопросы.

- Как выбрать начальные приближения  $x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}$ ?
- Каковы условия сходимости итерационного процесса?
- На какой итерации закончить вычисления?

Ответы на эти вопросы, совместно с расчетными соотношениями, и будут алгоритмом решения систем линейных уравнений методом итерации. Ответим на поставленные вопросы для случая линейных систем алгебраических уравнений.

### 7.1.1. Выбор начальных приближений

Если области, в которых находятся неизвестные  $x_i, i = 1, 2, \dots, n$ , известны, то начальные значения выбираются произвольно из

этой области. Если это невозможно, то за начальные приближения можно взять свободные члены  $\frac{b_1}{a_{11}}, \frac{b_2}{a_{22}}, \dots, \frac{b_n}{a_{nn}}$ .

## 7.1.2. Условия сходимости итерационного процесса

Условием сходимости итерационного процесса является правило, заключающееся в том, что сумма абсолютных значений отношений коэффициентов в каждом уравнении системы к диагональному должна быть меньше единицы.

Обеспечить сходимость итерационного процесса можно путем преобразования исходной системы к системе, которая будет эквивалентна исходной. Эти преобразования можно выполнить путем перестановки уравнений, выполнения над ними операций сложения и вычисления, а также умножения на постоянный коэффициент.

*Пример 7.2.*

Необходимо решить методом итераций следующую систему уравнений:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 &= 1; \\ -7x_1 - 2x_2 + 4x_3 &= 6; \\ 8x_1 + x_2 - 3x_3 &= 12. \end{aligned}$$

Эта система имеет решение:  $x_1 = 5, x_2 = -\frac{11}{2}, x_3 = \frac{15}{2}$ .

Однако решать эту систему уравнений методом итераций опасно, так как здесь не обеспечены условия сходимости итераций. Действительно, в первом уравнении  $\left|\frac{3}{2}\right| + \left|\frac{1}{2}\right| > 1$ , во втором  $\left|\frac{7}{2}\right| + \left|\frac{4}{2}\right| > 1$ ,

в третьем  $\left|\frac{8}{3}\right| + \left|\frac{1}{3}\right| > 1$ .

Для обеспечения условий сходимости преобразуем исходную систему уравнений. Второе уравнение поставим в первую строку.

Тогда  $\left| \frac{2}{7} \right| + \left| \frac{4}{7} \right| < 1$ . Заметим, что первое уравнение можно заменить также третьим.

Умножим первое уравнение на 4 и сложим его со вторым. В результате получим:  $+x_1 + 10x_2 + 8x_3 = 4$ . Теперь  $\left| \frac{1}{10} \right| + \left| \frac{8}{10} \right| < 1$ .

Это уравнение можно сделать вторым.

Для получения третьего уравнения выполним следующие преобразования:

1. Сложим все уравнения.
2. Полученное на первом шаге уравнение умножим на 2.
3. Полученное на втором шаге уравнение сложим со вторым.

В результате получим:  $-x_1 + 2x_2 + 8x_3 = 44$ . Теперь  $\left| \frac{1}{8} \right| + \left| \frac{2}{8} \right| < 1$ ,

условие сходимости итераций выполнено. В результате всех этих операций будет получена следующая эквивалентная система уравнений:

$$-7x_1 - 2x_2 + 4x_3 = 6;$$

$$x_1 + 10x_2 + 8x_3 = 10;$$

$$-x_1 + 2x_2 + 8x_3 = 44.$$

Теперь условия сходимости итераций выполнены полностью. Решение системы уравнений итерационным методом возможно при следующих начальных условиях:

$$x_1^{(0)} = -\frac{6}{7}, \quad x_2^{(0)} = 1, \quad x_3^{(0)} = \frac{44}{8} = 5,5.$$



### 7.1.3. Признак окончания вычислений

Признаком окончания итерационного процесса, исходя из условий точности, можно в первом приближении считать условие:

$$\left| x_k^{(v+1)} - x_k^{(v)} \right| \leq \varepsilon,$$

где  $x_k^{(v+1)}, x_k^{(v)}$  — значение  $k$ -го неизвестного на  $(v+1)$  и  $(v)$  итерациях;  $\varepsilon$  — допустимая погрешность вычисления неизвестных.

В системе Derive 5 пользователь не задает допустимую погрешность. Программа выдает решение на каждой итерации, число которых задает пользователь. По результатам таблицы приближений выбирается значение неизвестных.

Существует два способа решения уравнений методом итераций:

- ☐ метод простой итерации;
- ☐ метод Зейделя.

Пользователь не имеет возможности выбирать метод итераций, так как функции и команды системы Derive 5 не различают этих методов. Метод решения уравнений в системах символьной математики для пользователя также неизвестен, следовательно, выбора не существует.

### 7.1.4. Алгоритмы метода итерации

Алгоритм метода простой итерации состоит из совокупности условий выбора начальных приближений, расчетных соотношений и признака окончания вычислений.

Условия выбора начальных приближений следующие:

$$x_1^0 = \frac{b_1}{a_{11}}, x_2^0 = \frac{b_2}{a_{22}}, \dots, x_n^0 = \frac{b_n}{a_{nn}}.$$

Расчетные соотношения:

$$x_1^{(v+1)} = \varphi_1(x_1^v, x_2^v, \dots, x_n^v);$$

$$x_2^{(v+1)} = \varphi_2(x_1^v, x_2^v, \dots, x_n^v);$$

...

$$x_n^{(v+1)} = \varphi_n(x_1^v, x_2^v, \dots, x_n^v).$$

Признак окончания вычислений:

$$|x_k^{v+1} - x_k^v| \leq \varepsilon,$$

где  $v$  — номер итерации,  $x_k^v$  — значение неизвестного на  $v$ -й итерации,  $\varepsilon$  — погрешность.

В алгоритме простой итерации все значения неизвестных на шаге  $(v+1)$  вычисляются по их значениям на предыдущем шаге  $v$ . В алгоритме Зейделя результаты вычисления  $x_1$  на шаге  $(v+1)$  используются для вычисления  $x_2, x_3, \dots, x_n$  на этом же шаге; результаты вычисления  $x_2$  на шаге  $(v+1)$  используются для вычисления неизвестных  $x_3, x_4, \dots, x_n$ , на этом же шаге и т. д. Расчетные соотношения имеют вид:

$$x_1^{(v+1)} = \varphi_1(x_1^v, x_2^v, \dots, x_n^v);$$

$$x_2^{(v+1)} = \varphi_2(x_1^{(v+1)}, x_2^v, \dots, x_n^v);$$

...

$$x_n^{(v+1)} = \varphi_n(x_1^{(v+1)}, x_2^{(v+1)}, \dots, x_{(n-1)}^{(v+1)}, x_n^v).$$

Очевидно, что алгоритм Зейделя позволяет получить решение с большей точностью, чем алгоритм простой итерации при том же числе итераций.

## Сравнительная оценка точных и итерационных методов

В методе простой итерации на одну итерацию необходимо выполнить приблизительно  $2n^2$  арифметических операций типа сложения и умножения, в то время как по методу Гаусса для

решения системы  $n$  уравнений необходимо выполнить  $\frac{2}{3}n^3$  операций. Тогда очевидно, что метод итераций более целесообразный для случая, когда возможно получить решение задачи не более чем за  $\frac{1}{3}n$  итераций. Другими словами, он становится выгодным при решении уравнений больших размерностей.

Логическая схема итерационных методов очень проста, поэтому компьютерные программы короче, чем в методе Гаусса.

Итерационные методы позволяют распараллеливать алгоритм, что дает возможность эффективно решать уравнения на многопроцессорных компьютерах.

Недостатки итерационных методов заключаются в том, что они требуют от пользователя проверки условий сходимости итераций и, если они не выполняются, преобразования исходных уравнений к виду, когда обеспечивается сходимость итерационного процесса. Кроме этого, итерационные методы требуют выбора начальных приближений. Все это существенно усложняет компьютерные технологии решения уравнений.

## 7.2. Решение систем линейных уравнений с помощью системы Derive 5

### 7.2.1. Традиционные методы решения системы уравнений

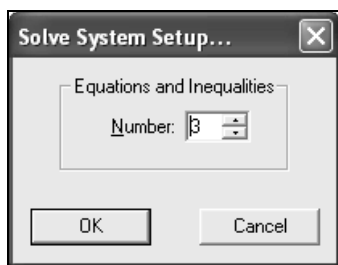
Системы линейных алгебраических уравнений в системе Derive 5 решаются с помощью функции `SOLVE`, имеющей вид:

```
Solve ( [ f1 (x1, x2, ..., xn) , f2 (x1, x2, ..., xn) , ..., fn (x1, x2, ..., xn) ,  
[x1, x2, ..., xn] ) ;
```

где  $f_i(x_1, x_2, \dots, x_n)$ ,  $i=1, 2, \dots, n$  —  $i$ -е уравнение системы,  $x_i$  — неизвестное, которое необходимо определить ( $i=1, 2, \dots, n$ ).

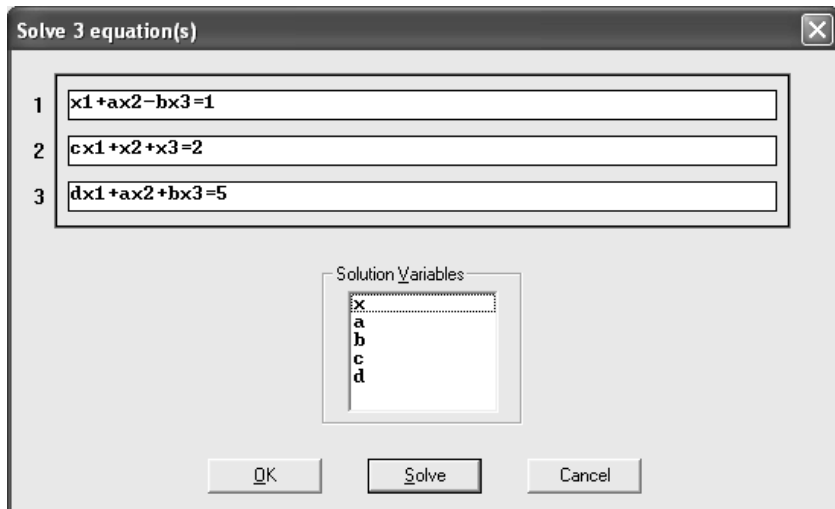
Технология решения систем уравнений заключается в выполнении следующих действий:

- ❑ щелчок мышью по пункту главного меню **Solve | System** на экране формируется окно **Solve System Setup** (рис. 7.1);



**Рис. 7.1.** Окно установки размерности системы уравнений

- ❑ установка в области ввода **Number** размерности системы уравнений (число уравнений) и щелчок по кнопке **OK** (на экране формируется новое окно **Solve 3 equation(s)**, показанное на рис. 7.2, с пустыми строками вектора уравнений);



**Рис. 7.2.** Окно ввода систем уравнений

- ввод последовательно всех уравнений (для перемещения курсора по строкам таблицы используются клавиши табуляции); на панели **Solution Variables** указать все искомые неизвестные (последовательно щелкать мышью по именам переменным, которые указаны в окне данной панели);
- щелкнуть мышью по кнопке **Solve** или **OK**, при этом в первом случае на экране появится функция и значения неизвестных, во втором — только функция и для ее исполнения необходимо нажать кнопку **Simplify** или **Approximate**, расположенные на панели инструментов.

Описанную выше технологию рассмотрим на примерах решения систем уравнений.

### Пример 7.3.

Пусть необходимо решить следующую систему уравнений:

$$ax_1 + bx_2 = 3;$$

$$2x_1 - x_2 = -1.$$

Прежде чем решать эту систему уравнений, настроим Derive 5 на прием переменных с индексами (пункт главного меню **Declare | Input | Word**). На экране появится сообщение:

```
#1: InputMode := Word.
```

Технология решения уравнений состоит в выполнении перечисленных ниже действий.

- Щелчок мышью по пункту главного меню **Solve**. В появившемся всплывающем меню щелкнуть мышью по пункту **System**. В результате выполненных действий на экране появится окно **Solve System Setup**, которое показано на рис. 7.1.
- Ввод в поле **Number** размерности системы (в нашем случае это цифра 3). После этого требуется нажать кнопку **OK** (на экране появится новое окно — **Solve 3 equation(s)**, которое показано на рис. 7.2).
- Ввод уравнений.
- Щелчок мышью в поле списка, расположенного на панели **Solutions Variables**. В поле списка появятся переменные  $a$ ,  $b$ ,  $x_1$ ,  $x_2$ .

- В заключение необходимо щелкнуть мышью по искомым неизвестным  $x_1, x_2$  и нажать кнопку **OK**.

В результате проделанных действий на экране появится функция **SOLVE** и значения неизвестных в виде:

#1: **InputMode := Word**

#2: **SOLVE**( $[a \cdot x_1 + b \cdot x_2 = 3, 2 \cdot x_1 - x_2 = -1], [x_1, x_2]$ )

#3: 
$$\left[ x_1 = \frac{3 - b}{a + 2 \cdot b} \wedge x_2 = \frac{a + 6}{a + 2 \cdot b} \right]$$

#4:  $a \cdot x_1 + b \cdot x_2 = 3$

#5:  $2 \cdot x_1 - x_2 = -1$

#6: 
$$\left[ \begin{array}{l} a \cdot x_1 + b \cdot x_2 = 3 \\ 2 \cdot x_1 - x_2 = -1 \end{array} \right]$$

#7: **SOLVE** $\left(\left[ \begin{array}{l} a \cdot x_1 + b \cdot x_2 = 3 \\ 2 \cdot x_1 - x_2 = -1 \end{array} \right], [x_1, x_2] \right)$

#8: 
$$\left[ x_1 = \frac{3 - b}{a + 2 \cdot b} \wedge x_2 = \frac{a + 6}{a + 2 \cdot b} \right]$$

Приведенная технология решения систем уравнений имеет недостаток. При необходимости редактирования уравнений пользователь должен повторно вводить в окно **Solve 3 equation(s)** все уравнения, хотя редактировать необходимо только одно из них. Этого недостатка лишена технология, которая рассматривается ниже.

- Набор и ввод уравнений системы, при этом на экране формируются уравнения вида:

#1:  $a \cdot x_1 + b \cdot x_2 = 3;$

#2:  $2 \cdot x_1 - x_2 = -1.$

- Щелчок мышью по кнопке **Author Matrix**, расположенной на панели инструментов (на экране формируется окно **Matrix Setup**, показанное на рис. 7.3).

- В появившемся окне ввод в поле **Rows** числа уравнений и поле **Columns** количества столбцов (в нашем случае 2 и 1).

- Щелкнуть мышью по кнопке **OK**, в результате чего на экране появится новое окно — **Author 2 x 1 matrix**, показанное на рис. 7.4.

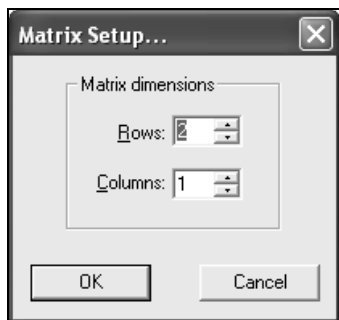


Рис. 7.3. Окно ввода размера системы уравнений

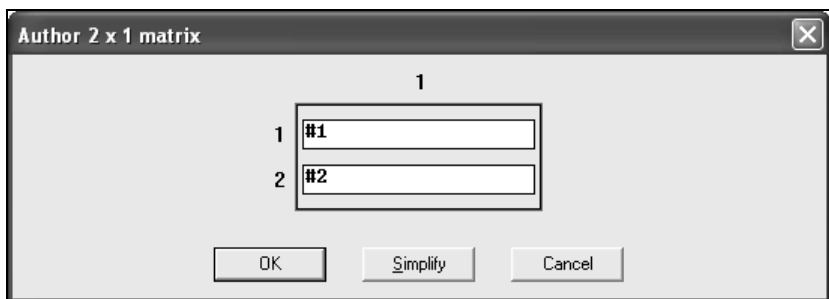


Рис. 7.4. Окно ввода уравнений

- ☐ Ввод в окне **Author 2 x 1 matrix** номеров строк экрана, в которых находятся уравнения (в нашем случае #1 и #2).
- ☐ Щелкнуть мышью по кнопке **OK** (на экране формируется система уравнений (#6)).
- ☐ Щелкнуть мышью по кнопке **Solve Expression**, расположенной на панели инструментов (на экране формируется окно **Solve Expression**).
- ☐ Выделить в окне **Solve Expression** неизвестные  $x_1$  и  $x_2$  (рис. 7.5).
- ☐ Щелкнуть мышью по кнопке **OK** (на экране формируется функция `SOLVE(#7)`).
- ☐ Щелкнуть мышью по кнопке **Simplify**, расположенной на панели инструментов (на экране формируется решение — #8).

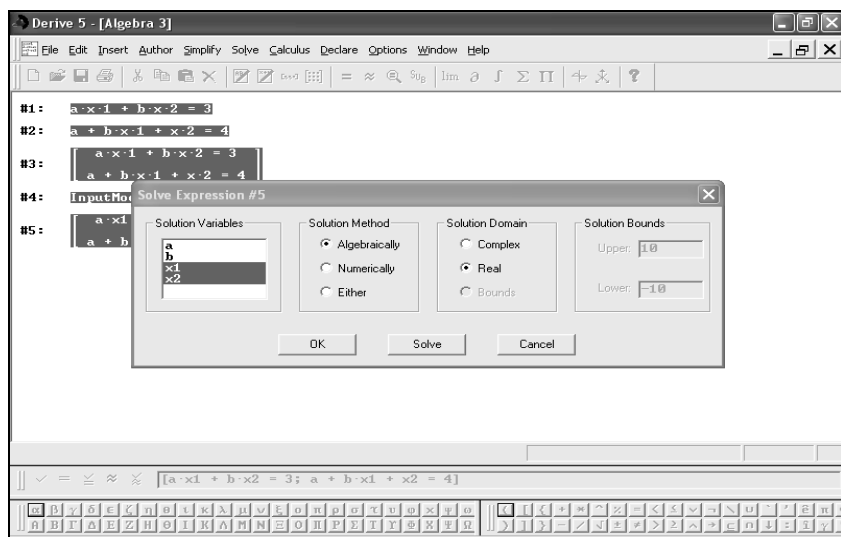


Рис. 7.5. Окно выбора вида ответа

*Пример 7.4.*

Решить три системы уравнений, приведенные в примере 4.1, первая является совместной и определенной, вторая — совместной и неопределенной, а третья — несовместной.

Решения, выполненные по приведенной выше технологии, имеют вид:

```
#1:  InputMode := Word
#2:  x1 + 2·x2 - x3 = 1
#3:  2·x1 + x2 + x3 = 2
#4:  3·x1 + 2·x2 + 3·x3 = 5
#5:  
$$\begin{bmatrix} x1 + 2 \cdot x2 - x3 = 1 \\ 2 \cdot x1 + x2 + x3 = 2 \\ 3 \cdot x1 + 2 \cdot x2 + 3 \cdot x3 = 5 \end{bmatrix}$$

#6:  SOLVE  $\begin{bmatrix} x1 + 2 \cdot x2 - x3 = 1 \\ 2 \cdot x1 + x2 + x3 = 2 \\ 3 \cdot x1 + 2 \cdot x2 + 3 \cdot x3 = 5 \end{bmatrix}, [x1, x2, x3], \text{Real}$ 
#7:   $[x1 = 0 \wedge x2 = 1 \wedge x3 = 1]$ 
```



$$\#8: \quad x_1 + 2 \cdot x_2 - x_3 = 1$$

$$\#9: \quad 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2$$

$$\#10: \quad 3 \cdot x_1 + 2 \cdot x_2 + 3 \cdot x_3 = 5$$

$$\#11: \quad \begin{bmatrix} x_1 + 2 \cdot x_2 - x_3 = 1 \\ 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2 \\ 3 \cdot x_1 + 2 \cdot x_2 + 3 \cdot x_3 = 5 \end{bmatrix}$$

$$\#12: \quad \text{SOLVE} \left( \begin{bmatrix} x_1 + 2 \cdot x_2 - x_3 = 1 \\ 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2 \\ 3 \cdot x_1 + 2 \cdot x_2 + 3 \cdot x_3 = 5 \end{bmatrix}, [x_1, x_2, x_3], \text{Real} \right)$$

$$\#13: \quad [x_1 + 2 \cdot x_3 = 2 \wedge 2 \cdot x_2 - 3 \cdot x_3 = -1]$$

$$\#14: \quad x_1 + 2 \cdot x_2 - x_3 = 1$$

$$\#15: \quad 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2$$

$$\#16: \quad 3 \cdot x_1 + 6 \cdot x_2 - 3 \cdot x_3 = 3$$

$$\#17: \quad \begin{bmatrix} x_1 + 2 \cdot x_2 - x_3 = 1 \\ 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2 \\ 3 \cdot x_1 + 6 \cdot x_2 - 3 \cdot x_3 = 3 \end{bmatrix}$$

$$\#18: \quad \text{SOLVE} \left( \begin{bmatrix} x_1 + 2 \cdot x_2 - x_3 = 1 \\ 2 \cdot x_1 + 4 \cdot x_2 - 2 \cdot x_3 = 2 \\ 3 \cdot x_1 + 6 \cdot x_2 - 3 \cdot x_3 = 3 \end{bmatrix}, [x_1, x_2, x_3], \text{Real} \right)$$

$$\#19: \quad [x_1 + 2 \cdot x_2 - x_3 = 1]$$

Из ответов (#7, #13, #19) видно, что первая система совместная и определенная, вторая совместная и неопределенная, третья — несовместная.

### Пример 7.5.

Необходимо решить следующую систему линейных уравнений:

$$x_1 + 2x_2 - 3,2x_3 + 3,4x_4 = 1,2;$$

$$2x_1 + x_2 - 0,2x_3 - 0,4x_4 = 1,8;$$

$$3x_1 + 1,5x_2 - 0,3x_3 - 0,6x_4 = 2,7;$$

$$-5x_1 - 10x_2 + 16x_3 - 17x_4 = 6.$$

Решение этой системы уравнений выполняется по описанной выше технологии.

$$\#1: \quad x_1 + 2 \cdot x_2 - 3.2 \cdot x_3 + 3.4 \cdot x_4 = 1.2$$

$$\#2: \quad 2 \cdot x_1 + x_2 - 0.2 \cdot x_3 - 0.4 \cdot x_4 = 1.8$$

$$\#3: \quad 3 \cdot x_1 + 1.5 \cdot x_2 - 0.3 \cdot x_3 - 0.6 \cdot x_4 = 2.7$$

$$\#4: \quad -5 \cdot x_1 - 10 \cdot x_2 + 16 \cdot x_3 - 17 \cdot x_4 = -6$$

$$\#5: \quad \begin{bmatrix} x_1 + 2 \cdot x_2 - 3.2 \cdot x_3 + 3.4 \cdot x_4 = 1.2 \\ 2 \cdot x_1 + x_2 - 0.2 \cdot x_3 - 0.4 \cdot x_4 = 1.8 \\ 3 \cdot x_1 + 1.5 \cdot x_2 - 0.3 \cdot x_3 - 0.6 \cdot x_4 = 2.7 \\ -5 \cdot x_1 - 10 \cdot x_2 + 16 \cdot x_3 - 17 \cdot x_4 = -6 \end{bmatrix}$$

$$\#6: \quad \text{SOLVE} \left( \begin{bmatrix} x_1 + 2 \cdot x_2 - 3.2 \cdot x_3 + 3.4 \cdot x_4 = 1.2 \\ 2 \cdot x_1 + x_2 - 0.2 \cdot x_3 - 0.4 \cdot x_4 = 1.8 \\ 3 \cdot x_1 + 1.5 \cdot x_2 - 0.3 \cdot x_3 - 0.6 \cdot x_4 = 2.7 \\ -5 \cdot x_1 - 10 \cdot x_2 + 16 \cdot x_3 - 17 \cdot x_4 = -6 \end{bmatrix}, [x_1, x_2, x_3, x_4], \text{Real} \right)$$

$$\#7: \quad [15 \cdot x_1 + 14 \cdot x_3 - 21 \cdot x_4 = 12 \wedge 15 \cdot x_2 - 31 \cdot x_3 + 36 \cdot x_4 = 3]$$

Из полученного решения видно, что программа не нашла решения системы указанных уравнений. В чем же причина? Мы знаем, что решение системы линейных алгебраических уравнений существует, если ее главный определитель не равен нулю. Проверим это условие.

Технология вычисления определителей описана в гл. 4. Используя ее, получим следующее решение:

$$\#1: \quad \begin{bmatrix} 1 & 2 & -3.2 & 3.4 \\ 2 & 1 & -0.2 & -0.4 \\ 3 & 1.5 & -0.3 & -0.6 \\ -5 & -10 & 16 & -17 \end{bmatrix}$$

$$\#2: \quad \text{DET} \begin{bmatrix} 1 & 2 & -3.2 & 3.4 \\ 2 & 1 & -0.2 & -0.4 \\ 3 & 1.5 & -0.3 & -0.6 \\ -5 & -10 & 16 & -17 \end{bmatrix}$$

$$\#3: \quad 0$$

Главный определитель равен нулю, т. е. система уравнений является несовместной, а поэтому и не имеет решения.

Недостатком приведенных выше технологий является необходимость ввода всех уравнений системы с численными значениями коэффициентов. Процесс ввода можно упростить, если первоначально ввести одно общее уравнение с символьными переменными, а затем образовать систему уравнений путем подстановки

в общее уравнение численных значений коэффициентов каждого из уравнений системы. Рассмотрим эту технологию, решая систему уравнений примера 7.5.

- ❑ Набор и ввод уравнения  $ax_1 + bx_2 + cx_3 + dx_4 = s$  (на экране формируется выражение —  $ax1+bx2+cx3+dx4=s$ ).
- ❑ Ввод уравнений системы в последовательности:
  - щелкнуть мышью по кнопке **Sub**, расположенной на панели инструментов (в результате этого на экране появляется окно **Substitute for variables**;

### Примечание

В области панели **Variables** указаны все переменные уравнения, а в области **New Value** — мигающий курсор.

- выделить переменную, например,  $a$ ;
  - ввод в области **New Value** ее значение из первого уравнения системы (цифра — 1);
  - активизировать переменную  $b$  и, щелкая по вкладке **New Value**, ввести цифру 2 и т. д.;
  - после ввода всех коэффициентов первого уравнения и значения  $s$  его правой части, необходимо нажать кнопку **OK** (на экране появляется первое уравнение системы с числовыми значениями коэффициентов);
  - выделить общее уравнение и вновь щелкнуть по кнопке **Sub**. Ввести коэффициенты второго уравнения и т. д. (на экране образуются четыре уравнения, которые еще не являются системой уравнений).
- ❑ Образование системы уравнений. Для этого необходимо щелкнуть по кнопке **Autot Matrix**. На экране появляется окно **Matrix Setup** с двумя вкладками. После этого:
- ввести размер матрицы  $4 \times 1$  (четыре уравнения (**Rows**) в одном столбце (**Columns**));

- нажать кнопку **OK** (на экране формируется новое окно **Author 4×1 matrix**, в котором:
  - ◇ ввести номера строк, в которых находятся уравнения;
  - ◇ щелкнуть мышью по кнопке **OK** (на экране формируется система уравнений).

□ Получение решения системы уравнений. Для этого:

- щелкнуть мышью по кнопке **Solve Expression** (на экране формируется уже знакомое окно **Solve Expression**);
- на панели **Solution Variables** выделить все неизвестные  $x_1, x_2, x_3, x_4$ ;
- на панели **Solution Method** устанавливаем флажок **Algebraically**, а во вставке **Solution Domain** — **Real** (действительное число);
- щелкнуть мышью по кнопке **OK** или **Solve**.

### Примечание

При нажатой кнопке **OK** на экране устанавливается функция **SOLVE**, тогда как при нажатой кнопке **Solve** на экране устанавливается функция **SOLVE** и решение.

Из ответа видно, что решение не получено, значения  $x_1, x_2, x_3, x_4$  — не определены. Процедуры этой технологии на экране будут иметь вид:

#1:  $a \cdot x_1 + b \cdot x_2 + c \cdot x_3 + d \cdot x_4 = s$

#2:  $1 \cdot x_1 + 2 \cdot x_2 + (-3.2) \cdot x_3 + 3.4 \cdot x_4 = 1.2$

#3:  $2 \cdot x_1 + 1 \cdot x_2 + (-0.2) \cdot x_3 + (-0.4) \cdot x_4 = 1.8$

#4:  $3 \cdot x_1 + 1.5 \cdot x_2 + (-0.3) \cdot x_3 + (-0.6) \cdot x_4 = 2.7$

#5:  $(-5) \cdot x_1 + (-10) \cdot x_2 + 16 \cdot x_3 + (-17) \cdot x_4 = -6$

#6: 
$$\begin{bmatrix} 1 \cdot x_1 + 2 \cdot x_2 + (-3.2) \cdot x_3 + 3.4 \cdot x_4 = 1.2 \\ 2 \cdot x_1 + 1 \cdot x_2 + (-0.2) \cdot x_3 + (-0.4) \cdot x_4 = 1.8 \\ 3 \cdot x_1 + 1.5 \cdot x_2 + (-0.3) \cdot x_3 + (-0.6) \cdot x_4 = 2.7 \\ (-5) \cdot x_1 + (-10) \cdot x_2 + 16 \cdot x_3 + (-17) \cdot x_4 = -6 \end{bmatrix}$$

#7: 
$$\text{SOLVE} \left[ \begin{bmatrix} 1 \cdot x_1 + 2 \cdot x_2 + (-3.2) \cdot x_3 + 3.4 \cdot x_4 = 1.2 \\ 2 \cdot x_1 + 1 \cdot x_2 + (-0.2) \cdot x_3 + (-0.4) \cdot x_4 = 1.8 \\ 3 \cdot x_1 + 1.5 \cdot x_2 + (-0.3) \cdot x_3 + (-0.6) \cdot x_4 = 2.7 \\ (-5) \cdot x_1 + (-10) \cdot x_2 + 16 \cdot x_3 + (-17) \cdot x_4 = -6 \end{bmatrix}, [x_1, x_2, x_3, x_4], \text{Real} \right]$$

#8:  $[15 \cdot x_1 + 14 \cdot x_3 - 21 \cdot x_4 = 12 \wedge 15 \cdot x_2 - 31 \cdot x_3 + 36 \cdot x_4 = 3]$

Система Derive 5 позволяет решать системы уравнений в символьных переменных практически любого порядка. Недостатком такого решения является большая сложность получаемых ответов. Покажем это на примере.

*Пример 7.6.*

Необходимо решить следующую систему уравнений:

$$a_1x_1 + b_1x_2 + c_1x_3 = s_1;$$

$$a_2x_1 + b_2x_2 + c_2x_3 = s_2;$$

$$a_3x_1 + b_3x_2 + c_3x_3 = s_3.$$

Процедуры решения на экране имеют вид:

$$\#1: a1 \cdot x1 + b1 \cdot x2 + c1 \cdot x3 = s1$$

$$\#2: a2 \cdot x1 + b2 \cdot x2 + c2 \cdot x3 = s2$$

$$\#3: a3 \cdot x1 + b3 \cdot x2 + c3 \cdot x3 = s3$$

$$\#4: \begin{bmatrix} a1 \cdot x1 + b1 \cdot x2 + c1 \cdot x3 = s1 \\ a2 \cdot x1 + b2 \cdot x2 + c2 \cdot x3 = s2 \\ a3 \cdot x1 + b3 \cdot x2 + c3 \cdot x3 = s3 \end{bmatrix}$$

$$\#5: \text{SOLVE} \left( \begin{bmatrix} a1 \cdot x1 + b1 \cdot x2 + c1 \cdot x3 = s1 \\ a2 \cdot x1 + b2 \cdot x2 + c2 \cdot x3 = s2 \\ a3 \cdot x1 + b3 \cdot x2 + c3 \cdot x3 = s3 \end{bmatrix}, [x1, x2, x3], \text{Real} \right)$$

$$\#6: \left[ x1 = \frac{b1 \cdot (c2 \cdot s3 - c3 \cdot s2) + b2 \cdot (c3 \cdot s1 - c1 \cdot s3) + b3 \cdot (c1 \cdot s2 - c2 \cdot s1)}{a1 \cdot (b2 \cdot c3 - b3 \cdot c2) + a2 \cdot (b3 \cdot c1 - b1 \cdot c3) + a3 \cdot (b1 \cdot c2 - b2 \cdot c1)} \wedge x2 = - \frac{a1 \cdot (c2 \cdot s3 - c3 \cdot s2) + a2 \cdot (c3 \cdot s1 - c1 \cdot s3) + a3 \cdot (c1 \cdot s2 - c2 \cdot s1)}{a1 \cdot (b2 \cdot c3 - b3 \cdot c2) + a2 \cdot (b3 \cdot c1 - b1 \cdot c3) + a3 \cdot (b1 \cdot c2 - b2 \cdot c1)} \wedge x3 = \frac{a1 \cdot (b2 \cdot s3 - b3 \cdot s2) + a2 \cdot (b3 \cdot s1 - b1 \cdot s3) + a3 \cdot (b1 \cdot s2 - b2 \cdot s1)}{a1 \cdot (b2 \cdot c3 - b3 \cdot c2) + a2 \cdot (b3 \cdot c1 - b1 \cdot c3) + a3 \cdot (b1 \cdot c2 - b2 \cdot c1)} \right]$$

Из полученного решения видно, что решается довольно простая система, однако она имеет решение, которое достаточно сложно для анализа.

## 7.2.2. Матричный метод решения систем линейных алгебраических уравнений

Эффективным методом решения систем линейных алгебраических уравнений является матричный метод. Сущность его состоит в следующем.

Пусть  $A$  — матрица коэффициентов системы уравнений,  $X$  — вектор неизвестных,  $B$  — вектор правых частей системы уравнений. Тогда решение системы уравнений в матричной форме будет иметь вид:

$$X = A^{-1}B.$$

*Пример 7.7.*

Решить матричным методом следующую систему уравнений:

$$2x_1 - 3x_2 = 1;$$

$$x_1 + 5x_2 = -4.$$

В данном случае:

$$A = \begin{vmatrix} 2 & -3 \\ 1 & 5 \end{vmatrix}, \quad B = [1, -4], \quad X = A^{-1}B.$$

Решением системы уравнений будет:

$$x_1 = -\frac{7}{13}, \quad x_2 = -\frac{9}{13}.$$

Технология решения системы уравнений в Derive 5:

- ❑ щелкнуть мышью по кнопке **Author** главного меню системы, а затем **Matrix** или по кнопке **Author Matrix**, расположенной на панели инструментов (на экране формируется окно **Matrix Setup**);
- ❑ в области **Matrix dimensions** ввести размерность матрицы — **Rows** (число строк); **Columns** (число столбцов) (в нашем случае две строки и два столбца);
- ❑ нажать кнопку **OK** (на экране формируется окно **Author 2×2 matrix**);
- ❑ заполнить пустые поля числами матрицы  $A$ , нажать кнопку **OK** или **Simplify** (на экране в строке #1 формируется матрица  $A$ );
- ❑ щелкнуть мышью по кнопке **Author Vector**, расположенной на панели инструментов (на экране формируется окно **Vector Setup**);

- ❑ набрать в области **Elements** число элементов вектора  $B$  (в нашем случае два элемента) и щелкнуть по кнопке **OK** (на экране формируется окно **Author 2 element Vector**);
- ❑ заполнить пустые поля числами вектора  $B$  (в нашем случае это — 1 и  $-4$ ) и после этого нажать кнопку **OK** (на экране формируется вектор: #2:  $[1, -4]$ );
- ❑ набрать в строке пользователя выражения:  $\#1^{(-1)} * \#2$  и нажать клавишу <Enter>, в результате чего на экране формируется выражение:

$$\#3: \begin{bmatrix} 2 & -3 \\ 1 & 5 \end{bmatrix}^{-1} [1 \quad -4];$$

- ❑ щелкнуть мышью по кнопке **Simplify**, расположенной на панели инструментов, в результате чего на экране формируется ответ в виде вектора:

$$\begin{bmatrix} -\frac{7}{9} & -\frac{9}{13} \end{bmatrix}.$$

При нажатии кнопки **Approximate**, расположенной на панели инструментов, на экране формируется ответ в виде следующего вектора:  $[-0.5384615, -0.6923076]$ .

Процедуры решения системы уравнений на экране имеют вид:

$$\#1: \begin{bmatrix} 2 & -3 \\ 1 & 5 \end{bmatrix}$$

$$\#2: [1, -4]$$

$$\#3: \begin{bmatrix} 2 & -3 \\ 1 & 5 \end{bmatrix}^{-1} \cdot [1, -4]$$

$$\#4: \left[ -\frac{7}{13}, -\frac{9}{13} \right]$$

$$\#5: [-0.5384615384, -0.6923076923]$$

Матричный способ позволяет успешно решать системы линейных алгебраических уравнений с символьными переменными.

*Пример 7.8.*

Пусть система уравнений имеет вид:

$$a_1x_1 + b_1x_2 + c_1x_3 = s_1;$$

$$a_2x_1 + b_2x_2 + c_2x_3 = s_2;$$

$$a_3x_1 + b_3x_2 + c_3x_3 = s_3.$$

Определим неизвестные  $x_1, x_2, x_3$ , решая систему уравнений матричным методом.

В данном случае матрица коэффициентов и вектор правых частей уравнений имеют вид:

$$A = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} \quad B = [s_1, \quad s_2, \quad s_3], \text{ а решение системы уравне-}$$

ний запишется в виде:

$$X = A^{-1}B.$$

Ответ будет тот же, что получен в примере 7.6.

Система Derive 5 позволяет матричным методом решать линейные алгебраические уравнения в случае, если их правые части образуют не вектор переменных, а целую матрицу. При этом получается множество решений, соответствующее размерности матрицы.

Для выполнения таких вычислений используется функция `ROW_REDUCE(A, B)`, где  $A$  — матрица коэффициентов системы уравнений,  $B$  — матрица свободных членов системы.

Технологию решения уравнений рассмотрим на примере.

*Пример 7.9.*

Пусть необходимо решить следующую систему уравнений:

$$2x_1 - 3x_2 + x_3 = d;$$

$$-3x_1 + x_2 + 7x_3 = c;$$

$$x_1 + 5x_2 + 2x_3 = d.$$



при этом свободные члены  $b, c, d$  образуют следующую матрицу:

$$\begin{bmatrix} 3 & 2 & -1 \end{bmatrix}$$

$$\begin{bmatrix} 6 & 4 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 7 & -5 \end{bmatrix}.$$

Необходимо найти неизвестные  $x_1, x_2, x_3$  при трех следующих вариантах значений свободных членов:  $b = 3, c = 6, d = 1$ ;  $b = 2, c = 4, d = 7$ ;  $b = -1, c = 2, d = -5$ .

Технология решения задачи заключается в выполнении следующих действий:

- ☐ набор и ввод матрицы коэффициентов (на экране в строке #1 формируется матрица  $A$ );
- ☐ набор и ввод матрицы свободных членов (на экране в строке #2 формируется матрица  $B$ );
- ☐ набор и ввод функции `ROW_REDUCE (#1, #2)`;
- ☐ щелчок мыши по кнопке **Simplify** или **Approximate** для получения решения.

На экране решение будет иметь вид:

$$\#1: \begin{bmatrix} 2 & -3 & 1 \\ -3 & 1 & 7 \\ 1 & 5 & 2 \end{bmatrix}$$

$$\#2: \begin{bmatrix} 3 & 2 & -1 \\ 6 & 4 & 2 \\ 1 & 7 & -5 \end{bmatrix}$$

$$\#3: \text{ROW\_REDUCE} \left( \left[ \begin{bmatrix} 2 & -3 & 1 \\ -3 & 1 & 7 \\ 1 & 5 & 2 \end{bmatrix}, \begin{bmatrix} 3 & 2 & -1 \\ 6 & 4 & 2 \\ 1 & 7 & -5 \end{bmatrix} \right] \right)$$

$$\#4: \begin{bmatrix} 1 & 0 & 0 & \frac{5}{11} & \frac{16}{11} & -\frac{15}{11} \\ 0 & 1 & 0 & -\frac{40}{121} & \frac{81}{121} & -\frac{78}{121} \\ 0 & 0 & 1 & \frac{133}{121} & \frac{133}{121} & -\frac{25}{121} \end{bmatrix}$$

Из ответа видно, что получены следующие три решения системы уравнений:

$$x_1 = \frac{5}{11}, x_2 = \frac{40}{121}, x_3 = \frac{133}{121};$$

$$x_1 = \frac{16}{11}, x_2 = \frac{81}{121}, x_3 = \frac{133}{121};$$

$$x_1 = \frac{15}{11}, x_2 = \frac{78}{121}, x_3 = \frac{25}{121}.$$

### 7.3. Итерационные методы решения систем уравнений

Система нелинейных уравнений может иметь любое конечное число решений, не иметь ни одного решения (системы не совместные) или иметь бесчисленное число решений. Все численные методы решения нелинейных уравнений являются методами последовательных приближений. Алгоритмы этих методов включают в себя:

- условия выбора начальных приближений;
- рекуррентные соотношения, по которым определяются приближения к искомому решению;
- признак окончания вычислений.

При этом, методы отличаются только видом рекуррентных соотношений (расчетных формул).

Решение системы нелинейных уравнений состоит из двух этапов:

1. Определение областей, в которых находятся решения и выбор начальных приближений из этих областей.
2. Уточнение решений и обеспечение заданной точности определения неизвестных.

Основная трудность в практической реализации первого этапа, без которого невозможно воспользоваться расчетными формулами определения неизвестных. К сожалению, не существует про-

стных формальных методов определения областей решений и начальных приближений. Достаточно просто их можно получить лишь для простых случаев, когда имеется система, состоящая из двух уравнений с двумя неизвестными.

В этом случае можно применить графический метод, воспользовавшись компьютерной графикой.

Рассмотрим этот способ на примере.

*Пример 7.10.*

Пусть требуется найти начальные приближения решения следующей системы уравнений:

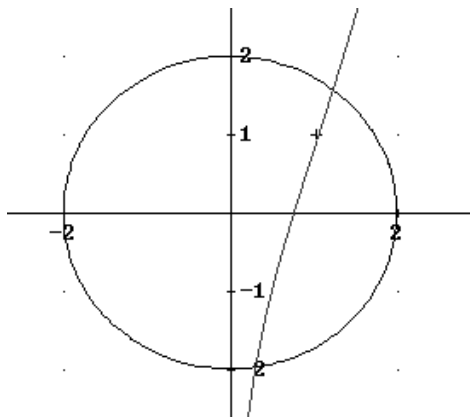
$$x_1^2 + x_2^2 - 4 = 0;$$

$$x_1^2 + 2 \ln x_1 - 1,2x_2 = 0.$$

Воспользуемся графической системой Derive 5.

Введем поочередно уравнения системы. На экране получим уравнения в строках #1 и #2. Первое уравнение является окружностью с радиусом  $R = 2$  и с центром в точке  $(0,0)$ . Второе уравнение является сложной кривой.

Графики функций показаны на рис. 7.6.



**Рис. 7.6.** Графическое представление функций

$$x_1^2 + x_2^2 - 4 = 0; x_1^2 + 2 \ln x_1 - 1,2x_2 = 0$$

Из рисунка видно, что имеются две точки пересечения кривых с координатами примерно равными  $(0,3; -2)$  и  $(1,2; 1,5)$ , которые и можно взять за начальные приближения.

При решении системы нелинейных уравнений с тремя неизвестными также можно воспользоваться графическим методом. Для этого необходимо построить множество функций при фиксированном значении одного из неизвестных. Из этого множества можно подобрать значения неизвестных, которые могут быть начальными приближениями.

В случае систем уравнений более высокого порядка графический способ вряд ли применим. Приходится подбирать начальные приближения методом проб и ошибок. Задача существенно облегчается, если пользователь знает ее физическую сущность. В этом случае он может из физических соображений указать диапазон параметров, в котором существует решение.

### Примечание

В системе Derive 5 реализованы два метода решения систем нелинейных уравнений: метод Ньютона и метод итераций.

## 7.4. Решение систем нелинейных уравнений

Системы нелинейных алгебраических уравнений в Derive 5 решаются методами Ньютона-Рафсона и итераций.

### 7.4.1. Метод Ньютона-Рафсона

### Примечание

В дальнейшем данный метод будет называться методом Ньютона.

Функция, реализующая метод Ньютона, имеет вид:

```
Newtons ([f1(x1, x2, ..., xn), f2(x1, x2, ..., xn),  
..., fn(x1, x2, ..., xn)], [x1, x2, ..., xn], [x10, x20, ..., xn0], n);
```

где  $f_i(x_1, x_2, \dots, x_n)$  —  $i$ -е уравнение системы,  $i = 1, 2, \dots, n$ ;  $x_i$  —  $i$ -е неизвестное;  $x_{i0}$  —  $i$ -е начальное приближение для неизвестного  $x_i$ ;  $n$  — число итераций, задаваемое пользователем.

Технологию решения нелинейных уравнений в среде Derive 5 рассмотрим на примере.

*Пример 7.11.*

$$xy + xz + yz = -19;$$

$$x^2 + y^2 + 2z^2 = 63;$$

$$\frac{x}{2y} + \frac{y}{2x} = \frac{13}{12}.$$

Необходимо решить методом Ньютона следующую систему нелинейных уравнений.

Начальными приближениями пусть будут следующие значения неизвестных:  $x_0 = -2$ ;  $y_0 = -1,5$ ;  $z_0 = 3$ .

Решить систему можно путем набора и ввода команды:

```
Newtons([x*y+x*z+y*z+19, x^2+y^2+2*z^2-63,
x/(2*y)+y/(2*x)-13/12], [x, y, z], [-2, -1.5, 3], 10).
```

После нажатия кнопки **Approximate** получим ответ в виде таблицы значений неизвестных на каждой итерации.

Однако на практике более рациональной может оказаться следующая компьютерная технология решения систем нелинейных уравнений:

□ ввод уравнений в любой последовательности, при этом на экране формируются следующие три уравнения:

$$\#1: xy + xz + yz + 19;$$

$$\#2: x^2 + y^2 + 2z^2 - 63;$$

$$\#3: \frac{x}{2y} + \frac{y}{2x} - \frac{13}{12};$$

- ввод функции `Newtons ([#1, #2, #3], [x, y, z], [-2, -1.5, 3])` (на экране формируется функция, в которой вместо номеров строк находятся уравнения);
- выполнение команды **Approximate** (на экране формируется решение системы уравнений в виде таблицы).

Процедуры решения системы уравнений и результаты на экране имеют следующий вид:

#1:  $x \cdot y + x \cdot z + y \cdot z + 19$

#2:  $\frac{x^2}{x} + \frac{y^2}{y} + \frac{z^2}{2 \cdot z} - 63$

#3:  $\frac{x}{2 \cdot y} + \frac{y}{2 \cdot x} - \frac{13}{12}$

#4: `NEWTONS`  $\left( \left[ x \cdot y + x \cdot z + y \cdot z + 19, \frac{x^2}{x} + \frac{y^2}{y} + \frac{z^2}{2 \cdot z} - 63, \frac{x}{2 \cdot y} + \frac{y}{2 \cdot x} - \frac{13}{12} \right], [x, y, z], [-2, -1.5, 3], 10 \right)$

#5:

	-2	-1.5	3
-2.526132404	-1.680313588	6.008710801	
-2.746119680	-1.830402562	5.175344152	
-2.952022783	-1.968522521	5.025556326	
-2.998042133	-1.998694357	5.001055883	
-2.999996347	-1.999997564	5.000001969	
-3	-2	5	
-3	-2	5	
-3	-2	5	
-3	-2	5	
-3	-2	5	

Из таблицы видно, что решением системы нелинейных уравнений является  $x = -3, y = -2, z = 5$ , при этом оно получено за шесть итераций.

## 7.4.2. Метод итераций

Функция решения систем нелинейных уравнений методом итераций имеет вид:

`Fixed_Point ([f1(x1,x2,...,xn), f2(x1,x2,...,xn), ..., fn(x1,x2,...,xn)], [x1,x2,...,xn], [x10,x20,...,xn0], n)`.

Сравнивая эту функцию с `Newtons`, несложно заметить, что они отличаются лишь названием. Процедуры решения систем уравнений те же, что и в методе Ньютона-Рафсона. В большинстве случаев метод Ньютона по сравнению с методом итераций позволяет получить решение за меньшее число итераций.

Метод итераций, при его практической реализации, требует преобразования уравнений и проверки условий сходимости итераций. Уравнения должны быть представлены в виде:

$$x = f_1(x, y, z);$$

$$y = f_2(x, y, z);$$

$$z = f_3(x, y, z).$$

При этом в функцию `Fixed_Point` входят лишь правые части уравнений.

Сходимость будет обеспечена, если выполнены условия сходимости итераций. В противном случае необходимо преобразовать каждое уравнение к виду, когда условия сходимости выполняются. Функции `Newtons` и `Fixed_Point` можно также применять для решения алгебраических и трансцендентных уравнений с одним неизвестным, а также систем линейных уравнений.

При решении уравнений с одним неизвестным функция `Newtons` имеет вид:

`Newtons ( [ f(x) ], [ x ], [ x0 ], n ),`

где  $f(x)$  — уравнение, представленное в виде функции (знак равно отсутствует);  $x$  — искомое неизвестное;  $x_0$  — начальное приближение;  $n$  — число итераций, задаваемое пользователем.

*Пример 7.12.*

Решить уравнение:

$$2^x - 4x = 0.$$

Процедуры решения уравнения имеют вид:

#1:  $2^x - 4 \cdot x$

#2: `NEWTONS([2x - 4·x], [x], [3], 8)`

#3:

3
5.588699449
4.710905999
4.191888106
4.017849970
4.000170826
4
4
4

#4: `NEWTONS([2x - 4·x], [x], [0], 8)`

#5:

0
0.3024023307
0.3099016185
0.3099069323
0.3099069323
0.3099069323
0.3099069323
0.3099069323
0.3099069323

Первый ответ (4) получен за шесть итераций. Второй (0.3099...) — за четыре итерации, при числе значащих цифр 10.

При решении уравнения с одним неизвестным методом итераций функция `Fixed_Point` имеет вид:

`Fixed_Point([f(x)], [x], [x0], n),`

где  $f(x)$  — функция, представляющая собой правую часть уравнения  $x = f(x)$ . В случае нашего примера уравнение  $2^x - 4x = 0$  будет представлено в виде:  $x = 2^{x-2}$  или  $x = 2 + \ln(x)/\ln(2)$ . Из условий сходимости итерационного процесса первое уравнение используется в функции `Fixed_Point` при определении корня из области изоляции  $[0—1]$ , второе — из области изоляции  $[3—5]$ .



Решение уравнения в среде Drive 5 имеет вид:

$$\#1: 2 + \frac{\text{LN}(x)}{\text{LN}(2)}$$

$$\#2: \text{FIXED\_POINT}\left(\left[2 + \frac{\text{LN}(x)}{\text{LN}(2)}\right], [x], [3], 16\right)$$

$$\begin{array}{c} 3 \\ 3.584962500 \\ 3.841958028 \\ 3.941841758 \\ 3.978869861 \\ 3.999997808 \\ 3.999999209 \\ 3.999999714 \\ 4 \end{array}$$

$$\#4: 2^{x-2}$$

$$\#5: \text{FIXED\_POINT}\left([2^{x-2}], [x], [0], 16\right)$$

$$\begin{array}{c} 0 \\ 0.25 \\ 0.297301787 \\ 0.307210998 \\ 0.309328356 \\ 0.3099069318 \\ 0.3099069322 \\ 0.3099069323 \\ 0.3099069323 \end{array}$$

Из сравнения результатов решения уравнения методами Ньютона и итераций следует, что метод Ньютона более производительный. Количество итераций для определения корня  $x = 4$  равно 6, а корня  $x = 0,309...$  равно 4. Число итераций по методу итераций равно соответственно 16 и 15.

Методы Ньютона и итераций могут успешно использоваться также для решения систем линейных уравнений.

Особенностями решения систем линейных уравнений методом Ньютона являются следующие:

- не нужно приводить уравнения к виду, когда итерационный процесс сходится;

- можно не указывать число итераций (решение получается на первой же итерации);
- диапазон начальных условий может быть очень широким;
- при нажатии кнопки **Simplify** система выдает точное решение;
- уравнение в функции **Newtons** представляется в виде выражения без знака равно (=).

Покажем это на примере.

*Пример 7.13.*

Пусть необходимо решить методами Ньютона и итераций следующую систему в среде Derive 5 уравнений:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 &= 1; \\ -7x_1 - 2x_2 + 4x_3 &= 6; \\ 8x_1 + x_2 - 3x_3 &= 12. \end{aligned}$$

Заметим, что уравнения представлены в виде, когда условия сходимости итераций не соблюдены.

Решение системы уравнений методом Ньютона при нулевых начальных условиях в среде Derive 5 имеет вид:

```
#1: 2·x1 + 3·x2 + x3 - 1
#2: - 7·x1 - 2·x2 + 4·x3 - 6
#3: 8·x1 + x2 - 3·x3 - 12
#4: NEWTONS([2·x1 + 3·x2 + x3 - 1, - 7·x1 - 2·x2 + 4·x3 - 6, 8·x1 + x2 - 3·x3 - 12], [x1,
x2, x3], [0, 0, 0], 8)
```

```
#5: 
$$\begin{bmatrix} 0 & 0 & 0 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \\ 5 & -5.5 & 7.5 \end{bmatrix}$$

```

Ответом является:  $x_1 = 5; x_2 = 5,5; x_3 = 7,5$  и решение получено за одну итерацию.

Решим теперь эту систему уравнений методом итераций, для чего представим ее в виде:

$$x_1 = -3/2x_2 - 1/2x_3 + 1/2;$$

$$x_2 = -7/2x_1 + 2x_3 - 3;$$

$$x_3 = 8/3x_1 + 1/3x_2 - 4.$$

Решение системы уравнений имеет вид:

$$\#1: -\frac{3}{2} \cdot x_2 - \frac{1}{2} \cdot x_3 + \frac{1}{2}$$

$$\#2: -\frac{7}{2} \cdot x_1 + 2 \cdot x_3 - 3$$

$$\#3: \frac{8}{3} \cdot x_1 + \frac{1}{3} \cdot x_3 - 4$$

$$\#4: \text{FIXED\_POINT} \left( \left[ -\frac{3}{2} \cdot x_2 - \frac{1}{2} \cdot x_3 + \frac{1}{2}, -\frac{7}{2} \cdot x_1 + 2 \cdot x_3 - 3, \frac{8}{3} \cdot x_1 + \frac{1}{3} \cdot x_3 - 4 \right], [x_1, x_2, x_3], [0, 0, 0], 8 \right)$$

$$\#5: \begin{bmatrix} 0 & 0 & 0 \\ 0.5 & -3 & -4 \\ 7 & -12.75 & -4 \\ 21.625 & -35.5 & 13.33333333 \\ 47.08333333 & -52.02083333 & 58.11111111 \\ 49.47569444 & -51.56944444 & 140.9259259 \\ 7.391203703 & 105.6869212 & 174.9104938 \\ -245.4856288 & 320.9517746 & 74.01337448 \\ -517.9343492 & 1004.226449 & -633.9572187 \end{bmatrix}$$

Из ответа видно, что решения нет и итерационный процесс расходится.

Преобразуем нашу систему к виду, когда условия сходимости итерационного процесса выполняются. Такая система будет иметь следующий вид:

$$x_1 = -2/7x_2 + 4/7x_3 - 6/7;$$

$$x_2 = -1/10x_1 - 8/10x_3 + 1;$$

$$x_3 = 1/8x_1 - 2/8x_2 + 44/8.$$

Решение методом итераций на экране имеет вид:

$$\#1: -\frac{2}{7} \cdot x_2 + \frac{4}{7} \cdot x_3 - \frac{6}{7}$$

$$\#2: -\frac{1}{10} \cdot x_1 - \frac{8}{10} \cdot x_3 + 1$$

$$\#3: \frac{1}{8} \cdot x_1 - \frac{2}{8} \cdot x_2 + \frac{44}{8}$$

$$\#4: \text{FIXED\_POINT} \left( \left[ \left[ -\frac{2}{7} \cdot x_2 + \frac{4}{7} \cdot x_3 - \frac{6}{7}, -\frac{1}{10} \cdot x_1 - \frac{8}{10} \cdot x_3 + 1, \frac{1}{8} \cdot x_1 - \frac{2}{8} \cdot x_2 + \frac{44}{8} \right], \right. \right. \\ \left. \left. [x_1, x_2, x_3], [0, 0, 0], 37 \right) \right]$$

0	0	0
-0.8571428571	1	5.5
2	-3.314285714	5.142857142
3.028571428	-3.314285714	6.578571428
3.848979591	-4.565714285	6.707142857
4.28	-4.750612244	7.122551020
4.999998928	-5.499998989	7.499999364
4.999999348	-5.499999384	7.499999613
4.999999603	-5.5	7.499999764
5	-5.5	7.5
5	-5.5	7.5
5	-5.5	7.5

Из решения видно, что ответ получен за 35 итераций (в методе Ньютона — одна итерация).

Ниже приводятся два задания для индивидуального решения систем уравнений с помощью Derive 5.

## Задание 1. Решение систем линейных алгебраических уравнений

Решить приведенные в табл. 7.1 системы уравнений матричным методом, методами Гаусса и Ньютона, а также методом итераций. Сравнить результаты этих методов, указав их достоинства и недостатки.

**Таблица 7.1.** Системы уравнений для задания 1

№ п/п	Система уравнений
1	$1,5x_1 - 0,8x_2 + 4,25x_3 = 5,1$ $1,2x_1 + 7,18x_2 - 3,2x_3 = 4,2$ $0,5x_1 - 1,5x_2 + 7,1x_3 = -1,2$

Таблица 7.1 (продолжение)

№ п/п	Система уравнений
2	$6,7x_1 - 0,6x_2 + 0,83x_3 = 6,8$ $0,8x_1 + 1,1x_2 + 7,2x_3 = 5,2$ $1,2x_1 + 5,4x_2 - 0,54x_3 = -3,2$
3	$-1,32x_1 + 2,15x_2 + 7,6x_3 = -1,4$ $2,62x_1 + 6,1x_2 - 4,12x_3 = 5,6$ $8,3x_1 - 2,84x_2 - 1,5x_3 = -6,5$
4	$0,51x_1 - 10,2x_2 - 3,26x_3 = -2,05$ $3,09x_1 + 1,23x_2 - 4,64x_3 = -5,6$ $3,2x_1 - 2,31x_2 - 8,4x_3 = 6,1$
5	$7,12x_1 - 6,66x_2 + 2,6x_3 = -3,1$ $-1,76x_1 + 6,5x_2 - 0,87x_3 = 2,85$ $0,65x_1 + 0,87x_2 - 8,7x_3 = 5,56$
6	$6,4x_1 - 0,73x_2 + 2,1x_3 = 3,8$ $-1,07x_1 + 3,8x_2 - 1,5x_3 = -1,2$ $2,7x_1 - 3,1x_2 + 4,2x_3 = -7,5$
7	$9,21x_1 - 1,84x_2 + 0,7x_3 = -3,2$ $-6,17x_1 + 8,5x_2 - 2,87x_3 = -3,75$ $0,7x_1 + 0,87x_2 - 8,7x_3 = 2,64$
8	$4,3x_1 - 1,2x_2 + 10,3x_3 = 4,2$ $0,21x_1 + 6,2x_2 + 3,54x_3 = 5,1$ $-0,31x_1 - 0,52x_2 + 3,6x_3 = -2,1$
9	$6,9x_1 + 2,3x_2 + 1,21x_3 = 3,1$ $x_1 + 2,3x_2 - 3,4x_3 = -2,3$ $0,21x_1 - 0,43x_2 + 6,3x_3 = 3,6$

Таблица 7.1 (продолжение)

№ п/п	Система уравнений
10	$12,4x_1 - 0,56x_2 + 4,2x_3 = 6,3$ $-0,65x_1 + 4,4x_2 + 1,5x_3 = 1,5$ $1,5x_1 + 2,1x_2 - 2,8x_3 = 1,7$
11	$1,2x_1 - 1,06x_2 - 6,7x_3 = 2,12$ $4,2x_1 - 6,3x_2 - 0,9x_3 = -1,1$ $0,6x_1 + 6,8x_2 + 0,82x_3 = 0,83$
12	$9,7x_1 + 0,35x_2 - 1,84x_3 = 2,15$ $4,64x_1 - 7,1x_2 - 4,3x_3 = 1,5$ $0,32x_1 + 0,348x_2 - 3,3x_3 = -3,1$
13	$6,5x_1 - 2,34x_2 + 1,4x_3 = 2,8$ $0,5x_1 + 7,3x_2 - 2,4x_3 = -3,8$ $8,6x_1 + 0,34x_2 - 6,4x_3 = 0,64$
14	$2,8x_1 + 4,3x_2 - 3,7x_3 = 5,1$ $-0,45x_1 - 8,24x_2 + 4,8x_3 = 5,4$ $0,54x_1 + 2,3x_2 + 3,7x_3 = 1,54$
15	$6x_1 + 0,13x_2 - 0,67x_3 = 1,9$ $3,8x_1 + 1,25x_2 - 4,3x_3 = 6,4$ $0,38x_1 - 0,64x_2 + 3,2x_3 = 5,4$
16	$1,5x_1 - 2,6x_2 + 7x_3 = -11,2$ $6,6x_1 + 1,3x_2 - 1,24x_3 = 5,3$ $0,85x_1 - 8,4x_2 + 4,7x_3 = 1,6$
17	$6,2x_1 - 0,52x_2 + 2,3x_3 = -1,8$ $-4,2x_1 + 3,4x_2 - 0,5x_3 = 0,7$ $0,2x_1 + 0,8x_2 + 3,6x_3 = 3,2$

Таблица 7.1 (окончание)

№ п/п	Система уравнений
18	$0,63x_1 - 0,54x_2 + 1,7x_3 = 3,6$ $0,65x_1 + 4,4x_2 + 0,15x_3 = 2,3$ $1,5x_1 + 0,2x_2 + 4,1x_3 = 2,8$
19	$8,4x_1 - 0,25x_2 + 3,1x_3 = -5,7$ $-0,3x_1 + 6,1x_2 - 1,54x_3 = 3,3$ $-6,8x_1 + 1,2x_2 - 7x_3 = 4,5$
20	$12x_1 + 4,2x_2 - 0,8x_3 = -5,4$ $-4,1x_1 + 2,2x_2 - 0,16x_3 = 1,6$ $-1,6x_1 - 4,3x_2 + 8,4x_3 = 12,2$

### Примечание

При решении системы уравнений методом итераций преобразуйте исходную систему к виду, когда итерационный процесс сходится. Решите задачу при точности определения неизвестных до 3, 10 и 20 знаков после запятой. Объясните результаты решения.

## Задание 2. Решение систем нелинейных алгебраических уравнений

Решить представленные в табл. 7.2 нелинейные системы уравнений методами Ньютона и итераций.

Таблица 7.2. Системы уравнений для задания 2

№ п/п	Система уравнений	Начальные приближения
1	$\sin(x_1 + x_2) - 1,2x_1 = 0,1$ $x_1^2 + x_2^2 = 1$	$x_1 = 0,74$ $x_2 = 0,67$
2	$\sin(x_2 + 1) - x_1 = 1,2$ $2x_2 + \cos x_1 = 2$	$x_1 = 0,74$ $x_2 = 0,67$

Таблица 7.2 (продолжение)

№ п/п	Система уравнений	Начальные приближения
3	$\operatorname{tg}(x_1 x_2 + 0,2) = x_1^2$ $0,6x_1^2 + 2x_2^2 = 1$	$x_1 = 0,88$ $x_2 = 0,52$
4	$\cos(x_2 - 1) + x_1 = 0,5$ $x_2 - \cos x_1 = 3$	$x_1 = 0,88$ $x_2 = 0,52$
5	$\sin x_1 + 2 \sin x_2 = 1$ $2 \sin 3x_1 + 3 \sin 3x_2 = 0,3$	$x_1 = 1,08$ $x_2 = 0,06$
6	$x_1^2 x_2 - x_2 - 9 = 0$ $x_1 x_2 - x_1^2 + 10 = 0$	$x_1 = 1,08$ $x_2 = 0,06$
7	$\operatorname{tg}(x_1 - x_2) - 4x_1 = 0$ $x_1^2 + 2x_2^2 = 1$	$x_1 = -0,5$ $x_2 = 0,6$
8	$x_1^2 + x_2^2 - x_1 x_2 = 0$ $\sin(x_1 + x_2) - 2,4x + 3,2 = 0$	$x_1 = -0,5$ $x_2 = 0,6$
9	$x_1^4 + x_2^2 - 3 = 0$ $x_1^3 + x_2^3 - 4 = 0$	$x_1 = 0,95$ $x_2 = 1,4$
10	$2x_1 x_2^2 - 4x_2 - 7,5 = 0$ $x_1^2 - 3x_1 x_2 + 4,5 = 0$	$x_1 = 0,95$ $x_2 = 1,4$
11	$\sin x_1 - x_2 = 1,3$ $\cos x_2 - x_1 = -0,82$	$x_1 = 1,8$ $x_2 = -0,35$
12	$\sin x + 2 \cos x - 0,8 = 0$ $x_1 x_2^2 + 3x_1 - 4,5 = 0$	$x_1 = 1,8$ $x_2 = -0,35$



Таблица 7.2 (окончание)

№ п/п	Система уравнений	Начальные приближения
13	$x_1^3 + x_2^3 - 6x_1 + 3 = 0$ $x_1^3 - x_2^3 - 6x_2 = 2$	$x_1 = 0,52$ $x_2 = -0,37$
14	$x_1^2 x_2^2 - x_2^2 - 18,75 = 0$ $x_1 + x_2^2 - 8,25 = 0$	$x_1 = 0,52$ $x_2 = -0,37$
15	$x_2 + e^{x_1 - x_2} = 0$ $x_1 + e^{x_1 + x_2} = 0$	$x_1 = -0,7$ $x_2 = -0,35$
16	$\operatorname{tg}(x_1 - x_2) - 4x_1 = 0$ $x_1^2 + 3x_2^3 - 4 = 0$	$x_1 = -0,7$ $x_2 = -0,35$
17	$x_1^2 x_2 - 8x_1 + 5,5 = 0$ $x_1 x_2 + 3x_2 - 10 = 0$	—
18	$x_1^2 \sin x_2 + x_2^2 \sin x_1 + 1 = 0$ $2x_1 + e^{(x_1 + x_2)} - 4 = 0$	—
19	$e^{x_1} + 2x_2 \ln x_1 - 3 = 0$ $x_1^2 x_2 - 3x_1 + 5,4 = 0$	—
20	$\sin x_1 + 3,5 \sin x_2 - 1 = 0$ $2 \sin 3x + 3 \sin 2x_2 - 0,4 = 0$	—

### Примечание

В уравнениях с нечетными номерами указаны значения начальных приближений (кроме № 17 и 19). В уравнениях с четными номерами начальные приближения должны быть определены учащимися. Если система уравнений второго порядка, то области начальных приближений легко найти графическим способом.



## Решение дифференциальных уравнений

### 8.1. Постановка задачи

Пусть дано дифференциальное уравнение первого порядка:

$$y' = f(x) \quad (8.1)$$

и начальное условие его решения  $y(x_0) = y_0$ .

Тогда решить уравнение — это значит найти такую функцию  $y = \varphi(x)$ , которая, будучи подставленной в исходное уравнение, обратит его в тождество и одновременно будет удовлетворено начальное условие. Задача отыскания функции  $y = \varphi(x)$  называется в математике задачей Коши.

При решении дифференциального уравнения порядка  $n$  задача Коши формулируется следующим образом.

Дано дифференциальное уравнение порядка  $n$ :

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}). \quad (8.2)$$

Необходимо найти такую функцию  $y = \varphi(x)$ , которая, будучи подставленной в исходное уравнение, обратит его в тождество и одновременно будут удовлетворены следующие  $n$  начальных условий:

$$\begin{aligned} y(x_0) &= y_0; \\ y'(x_0) &= y'_0; \end{aligned} \quad (8.3)$$

...

$$y^{(n-1)}(x_0) = y_0^{(n-1)}.$$

При численных методах решения дифференциального уравнения порядка  $n$  последнее обычно представляется в виде эквивалентной системы дифференциальных уравнений, каждое из которых является дифференциальным уравнением первого порядка, разрешенным относительно производной. Такое представление всегда возможно путем введения новых переменных.

Обозначим  $y' = y_1, y'_1 = y_2, \dots, y'_{(n-2)} = y_{n-1}$ . Тогда, подставляя новые переменные в исходное дифференциальное уравнение порядка  $n$ , получим следующую систему:

$$\begin{aligned} y' &= y_1; \\ y'_1 &= y_2; \\ &\dots \\ y'_{(n-2)} &= y_{n-1}; \\ y^{(n)} &= f(x, y, y_1, \dots, y_{(n-1)}). \end{aligned} \tag{8.4}$$

### Примечание

Систему (8.4) часто называют представлением уравнения порядка  $n$  в форме Коши.

Общим решением дифференциального уравнения (8.1) является семейство функций  $y = \varphi(x, C)$ , где  $C$  — произвольная постоянная, определяемая начальными условиями. Если удастся найти общее решение, то решение задачи Коши сводится к нахождению произвольной постоянной  $C$ . Однако в большинстве практических задач общее решение найти не удастся. Поэтому решать дифференциальные уравнения приходится приближенными методами.

Существуют две группы приближенных методов:

- *аналитические*, позволяющие получить приближенное решение в виде математического выражения;
- *численные*, позволяющие получить решение в виде таблицы.

## 8.2. Приближенные аналитические методы решения дифференциальных уравнений

### 8.2.1. Метод последовательного дифференцирования

Пусть заданы дифференциальное уравнение порядка  $n$  в виде выражения (8.2) и начальные условия в виде (8.3). Тогда можно доказать, что если правая часть уравнения (8.2) является аналитической функцией своих аргументов в окрестности точки  $(x_0, y_0, y'_0, \dots, y_0^{(n-1)})$ , то вблизи этой точки существует единственное решение задачи Коши, причем это решение можно разложить в ряд Тейлора. Отсюда следует, что конечный ряд Тейлора функции  $y = \varphi(x)$  является приближенным решением исходного дифференциального уравнения.

Ряд Тейлора функции  $y = \varphi(x)$  имеет вид:

$$\begin{aligned} y(x) = & y(x_0) + \frac{y'(x_0)}{1!}(x-x_0) + \frac{y''(x_0)}{2!}(x-x_0)^2 + \dots \\ & \dots + \frac{y^{(n-1)}(x_0)}{(n-1)!}(x-x_0)^{n-1} + \\ & + \frac{y^n(x_0)}{n!}(x-x_0)^n + \frac{y^{n+1}(x_0)}{(n+1)!}(x-x_0)^{n+1} + \dots \\ & \dots + \frac{y^k(x_0)}{k!}(x-x_0)^k + \dots \end{aligned} \quad (8.5)$$

Из выражения (8.5) видно, что  $n$  его первых членов  $y(x_0), y'(x_0), \dots, y^{(n-1)}(x_0)$ , известны и являются начальными условиями решения задачи. Это означает, что искомое приближенное решение дифференциального уравнения может быть записано в виде степенного ряда (степени  $n-1$ ), непосредственно по известным начальным условиям.

Для определения остальных членов степенного ряда (8.5) необходимо знать производные  $y^{(n)}(x_0), y^{(n+1)}(x_0), \dots, y^{(k)}(x_0), \dots$ .

Производную  $y^{(n)}(x_0)$  можно вычислить из исходного дифференциального уравнения (8.2), если подставить в его правую часть значения  $x, y, y', \dots, y^{(n-1)}$  из начальных условий, то есть

$x = x_0, y = y_0, y' = y'_0, \dots, y^{(n-1)} = y_0^{(n-1)}$ . Производные  $y^{(n+1)}(x_0), \dots, y^{(k)}(x_0), \dots$  можно вычислить путем последовательного дифференцирования исходного уравнения (8.2) и вычисления правых частей производных в точках  $x = x_0, y = y_0, y' = y'_0, \dots, y^{(n)} = y_0^{(n)}, \dots, y^{(k-1)} = y_0^{(k-1)}$ . Число членов степенного ряда (8.5) обычно определяется требуемой точностью решения дифференциального уравнения.

## 8.2.2. Метод неопределенных коэффициентов

Сущность метода рассмотрим на примере решения дифференциального уравнения второго порядка вида:

$$y'' + P(x)y' + Q(x)y = R(x) \quad (8.6)$$

при начальных условиях  $y(x_0) = y_0, y'(x_0) = y'_0$ .

Решение будем искать в виде степенного ряда:

$$y(x) = \sum_{n=0}^{\infty} c_n x^n. \quad (8.7)$$

Тогда неизвестными будут коэффициенты ряда  $c_n$ . Для их определения воспользуемся следующим приемом. Разложим в степенной ряд функции  $P(x), Q(x), R(x)$ :

$$P(x) = \sum_{n=0}^{\infty} p_n x^n,$$

$$Q(x) = \sum_{n=0}^{\infty} q_n x^n, \quad (8.8)$$

$$R(x) = \sum_{n=0}^{\infty} r_n x^n.$$

Продифференцируем дважды искомое решение (8.7):

$$y'(x) = \sum_{n=1}^{\infty} n c_n x^{n-1}, \quad (8.9)$$

$$y''(x) = \sum_{n=2}^{\infty} n(n-1) c_n x^{n-2}. \quad (8.10)$$

Подставим теперь (8.7), (8.8), (8.9), (8.10) в исходное уравнение (8.6). В результате получим:

$$\begin{aligned} & \sum_{n=2}^{\infty} n(n-1) c_n x^{n-2} + \sum_{n=0}^{\infty} p_n x^n \sum_{n=1}^{\infty} n c_n x^{n-1} + \\ & + \sum_{n=0}^{\infty} q_n x^n \sum_{n=0}^{\infty} c_n x^n = \sum_{n=0}^{\infty} r_n x^n. \end{aligned} \quad (8.11)$$

Уравнение (8.11) позволяет определить неизвестные коэффициенты  $c_n$ . Для этого достаточно приравнять коэффициенты при одинаковых степенях  $x$  и решить систему алгебраических уравнений. Заметим, что коэффициенты  $c_0$  и  $c_1$  при этом определяются из начальных условий. В выражении (8.11) находятся бесконечные степенные ряды. При решении практических задач ограничиваются определенным числом членов ряда, исходя из условий точности результатов.

Из описания метода неопределенных коэффициентов видно, что на порядок дифференциального уравнения ограничения не накладываются. Метод может быть использован при решении уравнений любого порядка.

### 8.2.3. Метод последовательных приближений

Пусть дано дифференциальное уравнение первого порядка вида (8.1) с теми же начальными условиями. Тогда приближение  $n$  к решению можно найти по следующей рекуррентной формуле:

$$y_n(x) = y_0(x) + \int_{x_0}^x f(x, y_{n-1}) dx, \quad (8.12)$$

где  $y_0(x)$  — начальное приближение;  $f(x, y_{n-1})$  — правая часть исходного дифференциального уравнения, полученная на  $(n-1)$  приближении.

В качестве начального приближения можно взять любую функцию, достаточно близкую к точному решению. Иногда, например, удобно в качестве начального приближения взять частичную сумму степенного ряда или даже начальное условие  $y(x_0)$ .

## 8.3. Аналитические методы решения дифференциальных уравнений в среде Derive 5

Система Derive 5 позволяет решать аналитическими методами системы дифференциальных уравнений (ДУ) первого и второго порядков. Для этого используются два библиотечных файла: `ode1.mth` и `ode2.mth`. Первый из них содержит функции решения ДУ первого порядка, второй — второго порядка. Особенности решения ДУ аналитическими методами в Derive 5 являются следующие:

- решение ДУ, в зависимости от его вида, получается в виде функции  $\varphi(x, y)$  или уравнения  $\varphi(x, y) = 0$ ;
- ответ может содержать интегралы, которые система не может "взять";
- решение может иметь специальные функции;

- ответ может быть представлен в вещественной и комплексной форме;
- если решение системой не получено, то откликом будет сообщение — "inapplicable" (означает — "неприменимо").

Для получения решения в явном виде  $y = f(x)$  необходимо представить решение в виде уравнения и разрешить его относительно  $y$ . После такого преобразования решение можно получить в графическом или табличном виде путем табулирования функции  $y = f(x)$ . При необходимости, специальные функции, содержащиеся в решении, можно разложить в степенной ряд, получив приближенное решение уравнения в аналитическом виде.

Система Derive 5 содержит несколько десятков функций решения ДУ. Рассмотрим только те, которые наиболее часто используются на практике.

### 8.3.1. Функции решения уравнений первого порядка

#### Функция **DSOLVE**

Эта функция позволяет решать уравнения первого порядка вида:

$$p(x, y) + q(x, y)y' = 0.$$

Представляется функция в одной из следующих форм:

- `DSOLVE1(p, q, x, y, x0, y0);`
- `DSOLVE1_GEN(p, q, x, y, c).`

Первая форма применяется для получения частного решения. Вторая — для получения общего решения. Символьные переменные в этих функциях имеют следующий смысл:

- $p, q$  — функции, содержащиеся в ДУ рассматриваемого вида;
- $x_0, y_0$  — начальные условия решения ДУ;
- $c$  — произвольная постоянная.

В дальнейшем решения ДУ аналитическими методами, с помощью функций Derive 5, будем рассматривать на примерах.



*Пример 8.1.* Пусть необходимо найти частное и общее решение уравнения:

$$\frac{x}{y} \frac{dy}{dx} + xy + 1 = 0$$

при начальных условиях  $y(1) = 1$ . В нашем примере  $p = xy + 1, q = \frac{x}{y}, x_0 = 1, y_0 = 1$ . Функции `DSOLVE` запишутся в следующем виде:

□ `DSOLVE1(xy+1, x/y, x, y, 1, 1);`

□ `DSOLVE1_GEN(xy+1, x/y, x, y, c).`

Технология решения задачи очень проста и состоит в выполнении следующих действий:

- набор и ввод одной из представленных функций;
- выполнение команды **Simplify** или **Approximate** (в первом случае ответ будет точным, во втором — численным с представлением чисел в естественной форме).

В результате решения будут получены следующие ответы:

□ частное решение —  $(1 - xy)/xy = \ln(x)$ ;

□ общее решение —  $1/xy = \ln(x) + c$ .

Ответ получен в виде уравнений. Для получения решения в явном виде их нужно разрешить относительно  $y$ . Ответ здесь очевиден:  $y = 1/x(\ln(x) + 1), y = 1/x(\ln(x) + c)$ .

Решение дифференциального уравнения показано на рис. 8.1 и 8.2.

На рис. 8.3 приведены результаты решения уравнения  $p(x, y) + q(x, y)y' = 0$  при следующих значениях  $p(x, y)$  и  $q(x, y)$ :

$$p = xy + 1, q = y/x, x_0 = 1, y_0 = 1;$$

$$p = x \ln(y) + 1 = x/y, x_0 = 1, y_0 = 1;$$

$$p = xy + 1, q = 2, x_0 = 1, y_0 = 1.$$

Из полученного решения видно, что в первом случае решения нет (ответ "inapplicable"), во втором случае в решении содержится не вычисленный интеграл и, наконец, в третьем случае решение получено в комплексной форме, содержащей специальную функцию.

#1: `LOAD(C:\DFWSTrial\DFW\MATH\ODE1.MTH)`

#2: `x·y + 1`

#3:  $\frac{x}{y}$

#4: `DSOLVE1`  $\left( x \cdot y + 1, \frac{x}{y}, x, y, 1, 1 \right)$

#5: 
$$\frac{1 - x \cdot y}{x \cdot y} = \text{LN}(x)$$

#6: `SOLVE`  $\left( \frac{1 - x \cdot y}{x \cdot y} = \text{LN}(x), y \right)$

#7: 
$$y = \frac{1}{x \cdot (\text{LN}(x) + 1)}$$

#8: `DSOLVE1_GEN`  $\left( x \cdot y + 1, \frac{x}{y}, x, y, c \right)$

**Рис. 8.1.** Решение дифференциального уравнения  
в аналитическом виде  
(получение первого ответа)

#9: 
$$\frac{1}{x \cdot y} = \text{LN}(x) + c$$

#10: `SOLVE`  $\left( \frac{1}{x \cdot y} = \text{LN}(x) + c, y \right)$

#11: 
$$y = \frac{1}{x \cdot (\text{LN}(x) + c)}$$

**Рис. 8.2.** Решение дифференциального уравнения  
в аналитическом виде (получение второго ответа)

```
#1:  LOAD(C:\DfWSTrial\DfW\MATH\ODE1.MTH)
#2:  DSOLVE1(x·y + 1,  y/x,  x,  y,  1,  1)
#3:                                     inapplicable
#4:  DSOLVE1(x·LN(y) + 1,  x/y,  x,  y,  1,  1)

#5:                                     
$$\hat{e}^x \cdot \text{LN}(y) + \int_1^x \frac{\hat{e}}{\hat{e}} d\hat{e} = 0$$

#6:  DSOLVE1(x·y + 1,  2,  x,  y,  1,  1)
#7:                                     
$$2 \cdot y \cdot \hat{e}^{x^2/4} + \sqrt{\pi} \cdot \hat{i} \cdot \left( \text{ERF}\left(\frac{\hat{i}}{2}\right) - \text{ERF}\left(\frac{\hat{i} \cdot x}{2}\right) \right) = 2 \cdot \hat{e}^{1/4}$$

```

**Рис. 8.3.** Решение дифференциального уравнения при различных значениях  $p(x, y)$  и  $q(x, y)$

## Функция **LINEAR1**

Функция решает ДУ вида:  $y' + p(x, y) = q(x)$  и имеет следующие две формы представления:

□ `LINEAR1(p, q, x, y, x0, y0);`

□ `LINEAR1_GEN(p, q, x, y, c).`

Первая функция дает частное, а вторая общее решение. Символьные переменные в этих функциях имеют тот же смысл, что и в функции `DSOLVE`.

*Пример 8.2.* Пусть необходимо решить уравнение  $dy(x)/dx + ay = 0$  при начальных условиях  $y(0) = 1$ .

В данном случае  $p = a, q = 0, x_0 = 0, y_0 = 1$ . Тогда функции будут иметь вид:

□ `LINEAR1(a, 0, x, y, 0, 1);`

□ `LINEAR1_GEN(a, 0, x, y, c).`

Решение уравнения показано на рис. 8.4.

В данном случае решение получено в явном виде.

#1: `LOAD(C:\DFW5Trial\DFW\MATH\ODE1.MTH)`

#2: `LINEAR1(a, 0, x, y, 0, 1)`

#3:  $y = \hat{e}^{-a \cdot x}$

#4: `LINEAR1_GEN(a, 0, x, y, c)`

#5:  $y = c \cdot \hat{e}^{-a \cdot x}$

**Рис. 8.4.** Решение уравнения  $dy(x)/dx + ay = 0$   
с использованием функции `LINEAR1`

*Пример 8.3.* Необходимо решить ДУ  $y' + 2xy = x$  (найти частное и общее решения), если начальными условиями являются:  $y(0) = 1$ .

В данном случае  $p = 2x, q = x, x_0 = 0, y_0 = 1$ . Тогда функции `LINEAR1` будут иметь вид:

□ `LINEAR1(2x, x, x, y, 0, 1);`

□ `LINEAR1_GEN(2x, x, x, y, c).`

Решение уравнения показано на рис. 8.5.

#1: `LINEAR1(2·x, x, x, y, 0, 1)`

#2:  $y = \frac{\hat{e}^{-x^2}}{2} + \frac{1}{2}$

#3: `LINEAR1_GEN(2·x, x, x, y, c)`

#4:  $y = c \cdot \hat{e}^{-x^2} + \frac{1}{2}$

**Рис. 8.5.** Решение уравнения  $y' + 2xy = x$   
с использованием функции `LINEAR1`

## Функция **SEPARABLE**

Функция решает ДУ вида:  $y' = p(x)q(y)$  и имеет следующие две формы представления:

□ `SEPARABLE(p, q, x, y, x0, y0);`

□ `SEPARABLE_GEN(p, q, x, y, c).`

### Примечание

Обозначения переменных в этой функции те же, что и в функциях, рассмотренных выше.

Функция позволяет определить частное и общее решение дифференциального уравнения при начальных условиях  $y(x_0) = y_0$ .

*Пример 8.4.* Дифференциальное уравнение имеет вид:  $y' = xy$ . Требуется найти частное и общее решения при начальных условиях  $y(0) = 1$ . В данном случае переменные функций `SEPARABLE` будут иметь значения:  $p = x, q = y, x_0 = 0, y_0 = 1$ , а функции запишутся в следующем виде:

□ `SEPARABLE(x, y, x, y, 0, 1);`

□ `SEPARABLE_GEN(x, y, x, y, c).`

Решим теперь более сложную задачу.

*Пример 8.5.* Пусть дифференциальное уравнение имеет вид:  $y' = (x^2 - 1)(y^2 + 1)$ , начальными условиями являются:  $y(0) = 1$ . В данном случае переменные функций будут иметь значения:  $p = x^2 - 1, q = y^2 + 1, x_0 = 0, y_0 = 1$ , а сами функции запишутся в следующем виде:

□ `SEPARABLE(x2-1, y2+1, x, y, 0, 1);`

□ `SEPARABLE_GEN(x2-1, y2+1, x, y, c).`

Откликом реализации этих функций будут решения уравнения в неявном виде.

## Функция **BERNOULLI**

Функция предназначена для решения дифференциального уравнения вида:  $y' + p(x)y = q(x)y^k$  и имеет следующие две формы представления:

□ `BERNOULLI(p, q, k, x, y, x0, y0);`

□ `BERNOULLI_GEN(p, q, k, x, y, c).`

**Примечание**

Обозначения переменных в этой функции те же, что и в функциях, рассмотренных выше.

Первая форма функции дает частное решение при заданных начальных условиях  $x_0$  и  $y_0$ . Вторая — общее решение.

*Пример 8.6.* Пусть необходимо решить следующее дифференциальное уравнение:  $y' + (x+1)y = xy^k$  при начальных условиях  $y(0)=1$  и значении  $k=2$ . В данном случае  $p = x+1, q = x$ . Тогда функции будут иметь вид:

□ `BERNOULLI(x+1, x, k, x, y, 0, 1);`

□ `BERNOULLI_GEN(x+1, x, k, x, y, c).`

После набора и ввода функций нажимается кнопка **Simplify** или **Approximate**, расположенные на панели инструментов. На экране будет сформировано аналитическое и численное решения уравнения при  $k=2$ .

### 8.3.2. Функции решения уравнений второго порядка

Решение дифференциальных уравнений второго порядка осуществляется функциями, которые содержатся в библиотечном файле `ode2.mth`. Технология решения уравнений практически не отличается от технологии решения уравнений первого порядка.

Ниже приводятся функции, которые позволяют наиболее просто решать уравнения второго порядка.

#### Функция **DSOLVE2**

Эта функция представляется в следующих трех формах:

□ `DSOLVE2(p, q, r, x, c1, c2);`

□ `DSOLVE2_BV(p, q, r, x, x0, y0, x1, y1);`

□ `DSOLVE2_IV(p, q, r, x, x0, y0, v0).`

Все формы функции предназначены для решения уравнения вида:  $y'' + p(x)y' + q(x)y = r(x)$ . По виду уравнения, символьным переменным функции `DSOLVE2` и примерам решения уравнений первого порядка понятно назначение символьных переменных в функциях `DSOLVE2`.

Рассмотрим примеры решения дифференциальных уравнений второго порядка указанного типа.

*Пример 8.7.* Пусть уравнение имеет вид:  $y'' + y' + y = x + 1$ . Необходимо решить дифференциальное уравнение с помощью приведенных выше функций при начальных условиях  $y(0) = 1$ ,  $y'(0) = 1$ . В данном примере символьные переменные функций имеют значения:  $p = 1$ ,  $q = 1$ ,  $r = x + 1$ ,  $x_0 = 0$ ,  $y_0 = 1$ ,  $x_1 = 0$ ,  $y_1 = 1$ .

Тогда функции решения дифференциального уравнения будут иметь вид:

- ❑ `DSOLVE2(1, 1, x+1, x, c1, c2);`
- ❑ `DSOLVE2_BV(1, 1, x+1, x, 0, 1, 0, 1);`
- ❑ `DSOLVE2_IV(1, 1, x+1, x, 0, 1, 0, 1).`

Обратим внимание на результаты решения по функциям `DSOLVE2_NV` и `DSOLVE_IV`. Параметры этих функций одинаковы, но ответы разные. Какое же из этих решений верное? Найдем первую и вторую производные и убедимся, что результат решения будет равен правой части исходного уравнения (то есть  $x + 1$ ) и будут удовлетворены начальные условия  $y(0) = 1$ ,  $y'(0) = 1$ . Определив первую и вторую производные с помощью команды **Differentiate**, расположенной на панели инструментов, и вычислив сумму  $y'' + y' + y$ , убеждаемся, что эта сумма равна  $x + 1$ . Однако начальные условия не соответствуют заданным и имеют значения:

- ❑ в случае функции `DSOLVE2_NV` —  $y(0) = 1$ ,  $y'(0) = 5$  (вместо  $y'(0) = 1$ );
- ❑ в случае функции `DSOLVE2_IV` —  $y(0) = 1$ ,  $y'(0) = 0$  (вместо  $y'(0) = 1$ ).

Таким образом, задача Коши не решена. Почему? В чем наша ошибка? Или функции не предназначены для решения задачи Коши?

## Функция **AUTONOMOUS\_CONSERVATIVE**

Функция находит решение дифференциального уравнения второго порядка  $y'' = f(x, y)$ . Она представляется в виде:

`AUTONOMOUS_CONSERVATIVE (q, x, y, x0, y0, y'0)`.

Символьные переменные функции имеют следующие значения:

□  $q$  —  $f(x, y)$ ;

□  $x, y$  — аргумент и искомая функция;

□  $x_0, y_0, y'0$  — начальные условия  $y(x_0) = y_0, y'(x_0) = y'_0$ .

*Пример 8.8.* Пусть требуется решить дифференциальное уравнение  $y'' = f(x, y)$ , если:

a)  $f(x, y) = y + 1$ ;

b)  $f(x, y) = x + 1$ ;

c)  $f(x, y) = x + y + 1$ .

Решение можно получить при нулевых начальных условиях, то есть  $y(0) = 0, y'(0) = 0$ .

В данном случае  $q_a = y + 1, q_b = x + 1, q_c = x + y + 1, x_0 = 0, y_0 = 0, y'_1 = 1$ .

Тогда функции будут иметь вид:

a) `AUTONOMOUS_CONSERVATIVE (y+1, x, y, 0, 0, 0)`;

b) `AUTONOMOUS_CONSERVATIVE (x+1, x, y, 0, 0, 0)`;

c) `AUTONOMOUS_CONSERVATIVE (x+y+1, x, y, 0, 0, 0)`.

После команды **Simplify** откликом будет следующее решение:

a)  $x = LN(\sqrt{y(y+2)}) + y + 1$ ;

b)  $x = \frac{\sqrt{2}\sqrt{y(x+1)}}{x+1}$ ;



$$c) \quad x = \pm LN(SIGN(x+1)) +$$

$$+ LN(\sqrt{y(2x+y+2)} + x + y + 1) - \frac{LN(x+1)^2}{2}.$$

## Функция *LILOUVILLE*

Функция предназначена для получения общего решения дифференциального уравнения вида:  $y'' + p(x)y' + q(y)(y')^2 = 0$ .

Функция имеет вид:

$$LILOUVILLE(p, q, x, y, c1, c2).$$

Обозначения переменных здесь аналогичны переменным, используемым в предыдущих функциях.

*Пример 8.9.* Пусть требуется найти общее решение дифференциального уравнения:  $y'' + y' + (y')^2 = 0$ .

В данном случае  $p=1, q=1$ . Тогда *LILOUVILLE* будет иметь вид:

$$LILOUVILLE(1, 1, x, y, c1, c2).$$

Откликом данной функции является следующее решение, в неявном виде:

$$-c_2 e^{-x} + e^y - c_1 = 0.$$

Проверим его правильность. Для этого представим ответ в явном виде, разрешив полученное уравнение относительно  $y$ . Найдя первую и вторую производные, подставим их в исходное дифференциальное уравнение и убедимся, что имеет место тождество.

Функция *LILOUVILLE* позволяет получить частное решение, если заданы начальные условия. Убедимся в этом на нашем примере. Пусть начальными условиями будут  $y(0)=1, y'(0)=1$ . Тогда функция будет иметь вид:

$$LILOUVILLE(1, 1, x, y, 0, 1, 1).$$

Откликом при реализации этой функции будет следующий ответ:

$e^y - e^{-x} = 0$  или в явном виде:  $y = -x$ . Подставив ответ в исходное дифференциальное уравнение, получим тождество  $0 = 0$ .

### 8.3.3. Решение дифференциальных уравнений в аналитическом виде

Аналитические методы решения дифференциальных уравнений дают решения в виде точных или приближенных формул. Это позволяет:

- представлять решения в виде графика;
- выполнять любые математические преобразования над полученным решением (дифференцировать, интегрировать, находить пределы, раскладывать в ряд и т. п.);
- изучать физическое явление, представляя полученную формулу как его математическую модель.

Покажем возможности аналитических методов на примерах исследования динамики систем массового обслуживания (СМО).

*Пример 8.10.* Функционирование одноканальной СМО с отказами описывается следующим дифференциальным уравнением:  $p'_0 + (\lambda + \mu)p_0 = \mu$ . Символы в уравнении имеют следующий смысл:

- $p_0$  — вероятность того, что СМО свободна от обслуживания (заявки отсутствуют);
- $\lambda$  — интенсивность потока заявок;
- $\mu$  — интенсивность обслуживания заявки.

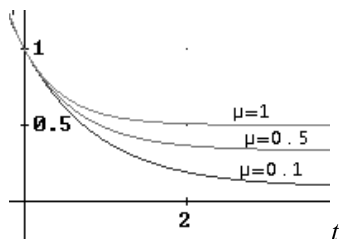
По структуре уравнения видно, что для его решения можно воспользоваться функцией `LINEAR1(p, q, x, y, x0, y0)`. В нашей задаче определяется функция  $p(t)$ , то есть переменными  $x, y$  будут соответственно —  $t, p$ . Предположим, что при  $t = 0$  СМО свободна от заявок. Тогда  $x_0 = t_0 = 0, y_0 = p(t_0) = 1$ . Подставляя переменные в выражение функции `LINEAR1`, получим:

`LINEAR1(λ+μ, μ, t, p, 0, 1)`.

После ввода этой функции и вызова команды **Simplify** получим частное решение в аналитическом виде. Упростив его с помощью пункта главного меню **Simplify** и функции **Expand**, получим ответ.

На рис. 8.6 приведены графики функции  $p = f(t)$  при интенсивности потока заявок  $\lambda=1$  [1/час] и различных значениях  $\mu$ . Из формулы и графиков можно сделать ряд важных выводов. Например:

- длительность переходных процессов в системе мала;
- установившееся значение вероятности свободного состояния системы равно  $\frac{\mu}{\lambda + \mu}$ ;
- если  $\lambda = \mu$ , то установившееся значение вероятности того, что система свободна от обслуживания равно 50%, то есть обслуживающий орган будет загружен лишь на половину рабочего времени и т. д.



**Рис. 8.6.** Графики функции  $p = f(t)$

*Пример 8.11.* Функционирование двухканальной системы массового обслуживания с отказами описывается следующим дифференциальным уравнением второго порядка:

$$p'' + (2\lambda + 3\mu)p' + (2\mu^2 + 2\lambda\mu + \lambda^2)p = 2\mu^2.$$

Как и раньше, символ  $p$  означает вероятность того, что система в произвольный момент времени  $t$  свободна от обслуживания, символы  $\lambda$  и  $\mu$  обозначают соответственно интенсивность потока заявок и интенсивность их обслуживания. Необходимо определить вероятность того, что в произвольный момент времени  $t$  СМО свободна от обслуживания заявок (заявки отсутствуют).

Будем считать, что вначале (при  $t = 0$ ) заявок на обслуживание нет. Тогда начальными условиями решения задачи будут:

$$p(0) = 1, p'(0) = 1.$$

Из структуры уравнения видно, что его решение можно получить с помощью функции

`DSOLVE2_BV(p, q, r, x, x0, y0, x1, y1)`.

В нашем случае  $p = (2\lambda + 3\mu)$ ,  $q = (2\mu^2 + 2\lambda\mu + \lambda^2)$ ,  $r = 2\mu^2$ ,  $x = t, x_0 = 0, y_0 = 1, x_1 = 0, y_1 = 1$ . Тогда функция `DSOLVE2_BV` будет иметь вид:

`DSOLVE_BV(2λ+3μ, 2μ²+2λμ+λ², 2μ², t, 0, 1, 0, 1)`.

Решение дифференциального уравнения в системе Derive 5 получить не сложно. После ввода функции `DSOLVE_BV` надо щелкнуть мышью по кнопке **Simplify** или **Approximate**, расположенных на панели инструментов. В результате на экране сформируется ответ. В первом случае в виде точного решения, во втором — в виде численного в естественной форме представления чисел. Полезно упростить ответ с помощью пункта **Simplify** и функции **Expand**. В конечном итоге получится следующее решение:

$$\frac{\lambda e^{\frac{t\sqrt{\mu(4\lambda+\mu)}}{2}} - \frac{t(2\lambda+3\mu)}{2}(\lambda+2\mu)}{\lambda^2 + 2\lambda\mu + 2\mu^2} + \frac{2\mu^2}{\lambda^2 + 2\lambda\mu + 2\mu^2}.$$

Решение в виде графика показано на рис. 8.7.

При решении дифференциальных уравнений необходимо выбирать только ту решающую функцию, которая соответствует виду решаемого уравнения. Если выбрать иную функцию, то получить решение можно, но оно может быть ошибочным или полученным с большой погрешностью. Примером является решение нашего уравнения с помощью функции `DSOLVE2_IV`. Эта функция отличается от функции `DSOLVE2_BV` только видом начальных условий (вместо переменных  $x_1, y_1$  пишется переменная  $v_0$ ). Функция:

`DSOLVE_IV(2λ+3μ, 2μ²+2λμ+λ², 2μ², t, 0, 1, 0, 1)`

выдает решение, отличное от полученного с помощью функции DSOLVE\_BV (#4, #5). Разница в результатах решения показанf на рис. 8.7.

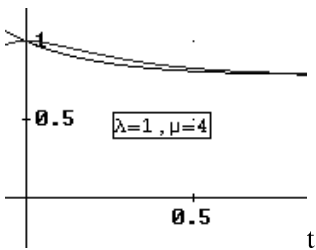


Рис. 8.7. Зависимость вероятности свободного состояния СМО от времени

# 8.4. Численные методы решения дифференциальных уравнений

## 8.4.1. Метод Эйлера

Пусть дано дифференциальное уравнение первого порядка  $y' = f(x, y)$  и начальные условия  $y(x_0) = y_0$ . Тогда численное решение уравнения по методу Эйлера находится в виде табл. 8.1.

Таблица 8.1. Таблица результатов численного решения дифференциального уравнения

$x_0$	$x_1$	$x_2$	...	$x_n$
$y_0$	$y_1$	$y_2$	...	$y_n$

Значения  $x_0, y_0$  являются начальными условиями решения уравнения, а поэтому не требуют вычислений. Первая строка таблицы, при известном  $x_0$  и шаге  $h$ , вычисляется по соотношениям:  $x_1 = x_0 + h, x_2 = x_1 + h, \dots, x_n = x_{n-1} + h$ . Значения функции  $y_1, y_2, \dots, y_n$  вычисляются по рекуррентной формуле Эйлера.

Получим эту формулу, для чего разложим функцию  $y = \varphi(x)$ , являющуюся решением исходного уравнения, в ряд Тейлора:

$$y(x) = y(x_0) + \frac{y'(x_0)}{1!}(x - x_0) + \frac{y''(x_0)}{2!}(x - x_0)^2 + \dots$$

$$\dots + \frac{y^{(n)}(x_0)}{n!}(x - x_0)^n.$$

Вычислим теперь значение функции в точке  $x = x_1 = x_0 + h$ :

$$y(x) = y(x_0) + \frac{y'(x_0)}{1!}(x_1 - x_0) + \frac{y''(x_0)}{2!}(x_1 - x_0)^2 + \dots$$

$$\dots + \frac{y^{(n)}(x_0)}{n!}(x_1 - x_0)^n.$$

Ограничиваясь первыми двумя членами разложения, получим:

$$y(x_1) = y(x_0) + \frac{y'(x_0)}{1!}(x_1 - x_0).$$

Так как  $x_1 - x_0 = h$ , а  $y'(x_0) = f(x_0, y_0)$ , то  $y(x_1) = y(x_0) + hf(x_0, y_0)$ . Принимая теперь  $x_1, y_1$  за начальные условия и проводя те же рассуждения, что при вычислении функции  $y(x_1)$ , получим:

$$y(x_2) = y(x_1) + hf(x_1, y_1);$$

$$y(x_3) = y(x_2) + hf(x_2, y_2);$$

...

$$y(x_n) = y(x_{n-1}) + hf(x_{n-1}, y_{n-1}).$$

Из этих выражений видно, что значения  $y_1, y_2, \dots, y_n$  табл. 8.1 могут быть вычислены по следующей рекуррентной формуле Эйлера:

$$y_{i+1} = y_i + hf(x_i, y_i). \quad (8.13)$$

Из вывода формулы (8.13) следует, что метод Эйлера основан на линейном разложении функции  $f(x, y)$  на участке  $h$  с последующей экстраполяцией решения за пределы этого участка.

Ошибка метода возникает за счет отбрасывания членов ряда Тейлора и имеет порядок  $h^2$ .

Метод Эйлера позволяет решать системы уравнений и дифференциальные уравнения высокого порядка. При решении системы уравнений формула Эйлера применяется на каждой итерации столько раз, сколько уравнений первого порядка содержится в системе. При решении дифференциального уравнения высокого порядка уравнение предварительно преобразуется в эквивалентную систему дифференциальных уравнений, каждое из которых первого порядка, а затем находится решение путем многократного использования формулы Эйлера.

### 8.4.2. Усовершенствованные методы Эйлера

Метод Эйлера дает большие погрешности, возрастающие от итерации к итерации. Усовершенствованные методы Эйлера дают решение с большей точностью. Все они относятся к так называемым методам прогноза и коррекции.

#### Метод Эйлера-Коши

По этому методу дифференциальное уравнение первого порядка вида  $y' = f(x, y)$  при начальных условиях  $y(x_0) = y_0$  решается по следующей рекуррентной формуле:

$$y_{i+1} = y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1})}{2}, \quad (8.14)$$

где  $y_{i+1}$  вычисляется по методу Эйлера, то есть  $y_{i+1} = y_i + hf(x_i, y_i)$ .

Точность усовершенствованного метода Эйлера-Коши выше метода Эйлера. Погрешность составляет порядка  $h^3$ .

#### Усовершенствованный метод Эйлера

По этому методу дифференциальное уравнение первого порядка решается по формуле:

$$y_{i+1} = y_i + hf\left(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}\right), \quad (8.15)$$

$$\text{где } x_{i+\frac{1}{2}} = x_i + \frac{h}{2}, \quad y_{i+\frac{1}{2}} = y_i + \frac{h}{2} f(x_i, y_i).$$

Погрешность этого метода такая же, как и усовершенствованного метода Эйлера-Коши.

### **Усовершенствованный метод Эйлера-Коши с итерационной обработкой результатов**

По этому методу дифференциальное уравнение первого порядка решается по формуле:

$$y_{i+1}^{(k)} = y_i + h \frac{f(x_i, y_i) + f(x_{i+1}, y_{i+1}^{(k-1)})}{2}, \quad k = 1, 2, \dots \quad (8.16)$$

В формуле (8.16)  $y_{i+1}^{(k-1)}$  является начальным приближением итерационного процесса и определяется по формуле Эйлера, то есть:

$$y_{i+1}^0 = y_i + hf(x_i, y_i).$$

Признаком окончания итерационного процесса является условие:

$$\left| y_{i+1}^{(k)} - y_{i+1}^{(k-1)} \right| \leq \varepsilon. \quad (8.17)$$

Из выражения (8.16) и условия (8.17) видно, что идея метода состоит в следующем. По формуле Эйлера (8.13) находится решение на  $(i+1)$ -м шаге, которое является на этом шаге первым приближением. Данное приближение используется для образования итерационного процесса на основе формулы усовершенствованного метода Эйлера-Коши. Признаком окончания итерационного процесса является условие (8.17). Итерационная обработка результатов вычисления по формуле (8.16) не позволяет найти точное решение дифференциального уравнения. Это объясняется тем, что формулы (8.13) и (8.16) содержат методические ошибки, которые не могут быть устранены итерационной обработкой результатов вычислений. Погрешность метода имеет порядок  $h^3$  на каждом шаге итераций.



### 8.4.3. Метод Рунге-Кутта

Метод Рунге-Кутта обладает более высокой точностью, чем методы Эйлера за счет снижения методических ошибок. Идея метода состоит в следующем.

По методу Эйлера решение дифференциального уравнения первого порядка определяется из соотношения:

$$y_{i+1} = y_i + \Delta y_i ;$$

где  $\Delta y_i = hf(x_i, y_i) = hy'(x_i, y_i)$ .

Тогда приращение  $\Delta y_i$  может быть найдено путем интегрирования:

$$\Delta y_i = \int_{x_i}^{x_{i+1}} y'(x, y) dx = \int_{x_i}^{x_{i+1}} f(x, y) dx .$$

Или окончательно

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx .$$

Вычислим теперь интеграл по методу прямоугольников:

$$y_{i+1} = y_i + (x_{i+1} - x_i) f(x_i, y_i) = y_i + hf(x_i, y_i) .$$

Из полученного выражения видно, что вычисление интеграла по методу прямоугольников приводит к формуле Эйлера.

Вычислим интеграл по формуле трапеций:

$$y_{i+1} = y_i + \frac{h}{2} (f(x_i, y_i) + f(x_{i+1}, y_{i+1})) .$$

Из выражения видно, что оно совпадает с расчетной формулой усовершенствованного метода Эйлера-Коши.

Для получения более точного решения дифференциального уравнения следует воспользоваться более точными методами вычисления интеграла.

В методе Рунге-Кутта искомый интеграл представляется в виде следующей конечной суммы:

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y) dx = y_i + \sum_{i=1}^q p_i K_i(h) , \quad (8.18)$$

где  $p_i$  — некоторые числа, зависящие от  $q$ ;  $K_i(h)$  — функции, зависящие от вида подынтегральной функции  $f(x, y)$  и шага интегрирования  $h$ , вычисляемые по следующим формулам:

$$K_1(h) = hf(x, y);$$

$$K_2(h) = hf(x + \alpha_2 h, y + \beta_{21} K_1(h));$$

$$K_3(h) = hf(x + \alpha_3 h, y + \beta_{31} K_1(h) + \beta_{32} K_2(h));$$

...

$$K_q(h) = hf(x + \alpha_q h, y + \beta_{q1} K_1(h) + \dots + \beta_{q,q-1} K_{q-1}(h)).$$

Значения  $p, \alpha, \beta$  получают из соображений высокой точности вычислений. Формулы Рунге-Кутты третьего порядка ( $q=3$ ) имеют следующий вид:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 4K_2 + K_3),$$

$$K_1 = hf(x_i, y_i),$$

$$K_2 = hf(x_i + \frac{h}{2}, y_i + \frac{K_1}{2}),$$

$$K_3 = hf(x_i + h, y_i + K_1 + 2K_2).$$

Наиболее часто используется метод Рунге-Кутты четвертого порядка, для которого расчетные формулы имеют следующий вид:

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4),$$

$$K_1 = hf(x_i, y_i),$$

$$K_2 = hf(x_i + \frac{h}{2}, y_i + \frac{K_1}{2}), \tag{8.19}$$

$$K_3 = hf(x_i + \frac{h}{2}, y_i + \frac{K_2}{2}),$$

$$K_4 = hf(x_i + h, y_i + K_3).$$

Формулы Рунге-Кутты имеют погрешности порядка  $h^{q+1}$ . Погрешность метода Рунге-Кутты четвертого порядка имеет порядок  $h^5$ .

## 8.5. Численные методы решения дифференциальных уравнений в среде Derive 5

Решение дифференциальных уравнений в Derive 5 осуществляется с помощью функций, находящихся в библиотечном файле `ode_appr.mth`. Файл содержит большое число различных функций решения таких уравнений. Рассмотрим основные из них.

### 8.5.1. Функция *EULER*

Функция предназначена для решения методом Эйлера дифференциального уравнения первого порядка  $y' = f(x, y)$  при начальных условиях  $y(x_0) = y_0$ . Функция представляется в следующем виде:

`EULER ( f ( x, y ) , x, y, x0, y0, h, n ) ,`

где:

- $f(x, y)$  — правая часть дифференциального уравнения, разрешенного относительно производной;
- $x, y$  — искомые неизвестные;
- $x_0, y_0$  — начальные условия;
- $h$  — шаг интегрирования;
- $n$  — число итераций.

Откликом является решение в виде таблицы с числом строк, равном  $n$ , и шагом  $h$ .

Технологию решения уравнений рассмотрим на примерах.

*Пример 8.12.* Пусть необходимо решить дифференциальное уравнение:

$$\frac{dy(x, y)}{dx} = \frac{x}{y} + 3e^{-2x},$$

при начальных условиях  $y(0) = 1$ . Выберем шаг интегрирования  $h = 0,2$  и число итераций  $n = 20$ .

В данном случае функция EULER будет иметь вид:

$EULER(x/y + 3e^{-2x}, x, y, 0, 1, 0.2, 20)$ .

Технология решения задачи будет следующей:

- ☐ обращение к библиотечному файлу ODE\_APPR.MTH — **File | Load | ODE\_APPR**;
- ☐ набор и ввод правой части исходного уравнения (на экране в строке #2 формируется выражение:  $x/y + 3 \cdot e^{(-2 \cdot x)}$ );
- ☐ набор и ввод функции  $EULER(\#2, x, y, 0, 1, 0.2, 20)$  (на экране формируется функция;
- ☐ щелчок мыши по кнопке **Approximate**, расположенной на панели инструментов (на экране формируется решение).

Процедуры решения уравнения показаны на рис. 8.8.

#1: `LOAD(C:\DFW5Trial\DFW\MATH\ODE_APPR.MTH)`

#2:  $\frac{x}{y} + 3 \cdot e^{-2 \cdot x}$

#3:  $EULER\left(\frac{x}{y} + 3 \cdot e^{-2 \cdot x}, x, y, 0, 1, 0.2, 20\right)$

0	1
0.2	1.6
0.4	2.027192027
0.6	2.336252860

#4: 

3.4	4.399477744
3.6	4.554709810
3.8	4.713235889
4	4.874784190

**Рис. 8.8.** Решение дифференциального уравнения методом Эйлера

## 8.5.2. Функция **TAY\_ODE1**

Функция выдает решение в виде многочлена степени  $n$ . Функция имеет следующий вид: `TAY_ODE1(f(x, y), x, y, x0, y0, n)`.

Обозначения в этой функции те же, что и в функции `EULER`, символ  $n$  обозначает степень полинома, получаемого в результате решения уравнения.

*Пример 8.13.* Пусть необходимо решить дифференциальное уравнение из предыдущего примера. В этом случае функция будет иметь вид:

`TAY_ODE1(x/y + 3·e-2x, x, y, 0, 1, n)`.

Решение получим в виде многочлена степени  $n=1,2,3$  путем редактирования функции в окне пользователя. Процедуры решения не нуждаются в комментариях. Они имеют вид, показанный на рис. 8.9.

The screenshot shows a software window with a list of steps on the left and a large text area on the right. The steps are as follows:

- #1:  $\frac{x}{y} + 3 \cdot e^{-2 \cdot x}$
- #2: `TAY_ODE1`  $\left( \frac{x}{y} + 3 \cdot e^{-2 \cdot x}, x, y, 0, 1, n \right)$
- #3: `TAY_ODE1`  $\left( \frac{x}{y} + 3 \cdot e^{-2 \cdot x}, x, y, 0, 1, 1 \right)$
- #4:  $3 \cdot x + 1$
- #5: `TAY_ODE1`  $\left( \frac{x}{y} + 3 \cdot e^{-2 \cdot x}, x, y, 0, 1, 2 \right)$
- #6:  $-\frac{5 \cdot x^2}{2} + 3 \cdot x + 1$
- #7: `TAY_ODE1`  $\left( \frac{x}{y} + 3 \cdot e^{-2 \cdot x}, x, y, 0, 1, 3 \right)$
- #8:  $\frac{3}{x} - \frac{5 \cdot x^2}{2} + 3 \cdot x + 1$

**Рис. 8.9.** Решение дифференциального уравнения с помощью функции `TAY_ODE1`

Ответами при  $n=1$ ,  $n=2$ ,  $n=3$  соответственно являются:

$$\square 3x+1;$$

$$\square -5/2x^2 + 3x + 1;$$

$$\square x^3 - 5/2x^2 + 3x + 1.$$

Пользуясь функцией `TAY_ODE1`, следует иметь в виду, что полученное решение в виде формулы пригодно в узком диапазоне аргумента, вблизи начальных условий.

### 8.5.3. Функция *PICARD*

Функция уточняет решение при известном приближении  $p(x)$ .

Она имеет вид:

$$\text{PICARD} (f(x, y), p, x, y, x0, y0).$$

Обозначения переменных здесь очевидны.

*Пример 8.14.* Пусть решением уравнения  $\frac{dy(x, y)}{dx} = \frac{x}{y} + 3e^{-2x}$

является полином первой степени  $3x+1$ , полученный в примере 8.13. Уточним решение. В данном случае функция `PICARD` будет иметь вид:

$$\text{PICARD} (x/y + 3e^{-2x}, 3x+1, x, y, 0, 1).$$

В результате решения уравнения получим следующий ответ, показанный на рис. 8.10.

$$\#1: \frac{x}{y} + 3 \cdot e^{-2 \cdot x}$$

$$\#2: \text{PICARD} \left( \frac{x}{y} + 3 \cdot e^{-2 \cdot x}, 3 \cdot x + 1, x, y, 0, 1 \right)$$

$$\#3: -\frac{3 \cdot e^{-2 \cdot x}}{2} - \frac{\text{LN}(3 \cdot x + 1)}{9} + \frac{x}{3} + \frac{5}{2}$$

Рис. 8.10. Использование функции `PICARD`

## 8.5.4. Функция *RK*

Система символьной математики Derive 5 содержит несколько десятков функций решения дифференциальных уравнений первого и второго порядков в символьном и числовом виде. Эти функции позволяют решать уравнения специального вида, с представлением решений в аналитическом виде, в виде чисел и числовых рядов. Наиболее важной является функция *RK*, которая дает решение дифференциальных уравнений и систем методом Рунге-Кутты.

Функция *RK* имеет вид:

*RK* (*r*, *u*, *u0*, *h*, *n*),

где:

- *r* — вектор правых частей уравнений, разрешенных относительно производных (при этом система уравнений представляется в виде, когда каждое из уравнений является уравнением первого порядка);
- *u* — вектор неизвестных;
- *u0* — вектор начальных условий;
- *h* — шаг интегрирования;
- *n* — число итераций.

### Примечание

Векторы представляются в виде квадратных скобок. Квадратные скобки опускаются, если решается дифференциальное уравнение первого порядка.

Откликом при реализации функции *RK* является матрица неизвестных с числом строк *n* и шагом *h*. Функция *RK* находится в файле *ODE\_APPR.mth*, к которому необходимо обратиться перед решением уравнений.

Технология решения дифференциальных уравнений с помощью этой функции следующая:

- обращение к файлу ODE\_APPR.mth — **File | Load | Utility | ODE\_APPR.mth**;
- ввод функции  $RK(r, u, u0, h, n)$  (на экране формируется функция);
- выполнение команды **Approximate** (на экране формируется ответ в виде матрицы).

При решении дифференциальных уравнений высокого порядка целесообразно, перед вводом функции  $RK$ , ввести поочередно правые части всех уравнений. После ввода они будут расположены на экране в отдельных строках, помеченных символами  $\#k$ , где  $\#k$  ( $k = 1, 2, 3, \dots, n$ ) номера строк, в которых находятся правые части всех  $n$  дифференциальных уравнений. Только после этого можно ввести функцию  $RK$ , записав вектор  $r$  в виде  $[\#1, \#2, \dots, \#n]$ .

*Пример 8.15.* Пусть необходимо решить следующее дифференциальное уравнение  $y'' + xy' - 2y + 1 = 0$  с начальными условиями  $y(0) = 1, y'(0) = 2$ .

Прежде чем обратиться к системе Derive 5, представим уравнение второго порядка в виде системы двух дифференциальных уравнений, каждое из которых первого порядка, разрешенное относительно производной. Такие преобразования выполняются методом подстановок. Обозначим  $y' = z$ . Тогда система дифференциальных уравнений будет иметь вид:

$$\begin{aligned} y' &= z; \\ z' &= -xz + 2y - 1. \end{aligned}$$

Теперь запишем функцию  $RK$ :

$RK([z, -xz + 2y - 1], [x, y, z], [0, 1, 2], 0.2, 8)$ .

После ввода этой функции и вызова команды **Approximate** получим ответ (рис. 8.11).



```

#1:  z
#2:  - x·z + 2·y - 1
#3:  RK([z, - x·z + 2·y - 1], [x, y, z], [0, 1, 2], 0.2, 10)
#4:

```

	0	1	2
0.2	1.422654	2.239849199	
0.4	1.901148032	2.557857621	
0.6	2.450708602	2.949502217	
0.8	3.085402653	3.407818874	
1	3.817721207	3.924150260	
1.2	4.658325186	4.488996498	
1.4	5.615960189	5.092832813	
1.6	6.697521979	5.726783792	
1.8	7.908236609	6.383088450	
2	9.251910923	7.055339819	

**Рис. 8.11.** Решение дифференциального уравнения с использованием функции RK

*Пример 8.16.* Пусть необходимо решить методом Рунге-Кутты следующую систему линейных дифференциальных уравнений:

$$\begin{aligned}
 \frac{dP_0(t)}{dt} &= -aP_0(t) + mP_1(t); \\
 \frac{dP_1(t)}{dt} &= aP_0 - (a + m)P_1(t) + 2mP_2(t); \\
 \frac{dP_2(t)}{dt} &= aP_1(t) - 2mP_2(t).
 \end{aligned}$$

Начальные условия решения задачи и переменные имеют следующие значения:  $P_0(0) = 1$ ,  $P_1(0) = 0$ ,  $P_2(0) = 0$ ,  $a = 0,001$ ,  $m = 3$ . Шаг интегрирования  $h = 0,1$ ; число итераций  $n = 50$ .

Заданная система дифференциальных уравнений описывает функционирование многоканальной системы массового обслуживания с отказами, имеющей два обслуживающих канала. В данном случае коэффициенты  $a$  и  $m$  являются переменными, поэтому запишем первоначально функцию RK для случая сим-вольных переменных.

Рациональной может быть следующая технология решения системы уравнений в среде Derive 5:

- ввод поочередно правых частей системы уравнений, при этом аргумент  $t$  опускается (на экране образуется три уравнения, например, в строках #1, #2, #3);
- ввод функции RK в виде —  $RK([ \#1, \#2, \#3 ], [ t, p_0, p_1, p_2 ], [ 0, 1, 0, 0 ], h, n)$  (на экране формируется функция RK с вектором правых частей системы дифференциальных уравнений);
- ввод с помощью команды **Sub**, вызываемой нажатием кнопки на панели инструментов, значений переменных  $a, m, h, n$  (на экране формируется функция RK с численными значениями переменных);
- выполнение команды **Approximate** (на экране формируется решение в виде матрицы с  $n$  строками и четырьмя столбцами  $t, P_0, P_1, P_2$ ).

Описанное выше решение показано на рис. 8.12.

```
#1: - a·p0 + n·p1
#2: a·p0 - (a + n)·p1 + 2·n·p2
#3: a·p1 - 2·n·p2
#4: RK([- a·p0 + n·p1, a·p0 - (a + n)·p1 + 2·n·p2, a·p1 - 2·n·p2], [t, p0, p1, p2], [0, 1, 0, 0], h, n)
#5: RK([- 0.01·p0 + 3·p1, 0.01·p0 - (0.01 + 3)·p1 + 2·3·p2, 0.01·p1 - 2·3·p2], [t, p0, p1, p2], [0, 1, 0, 0], 0.1, 50)
```

	0	1	0	0	
0.1	0.9991365011	0.0008631227998	3.760417916	10 <sup>-7</sup>	
0.2	0.9984972676	0.001501599360	1.132948609	10 <sup>-6</sup>	
0.3	0.9980239625	0.001974081500	1.955962748	10 <sup>-6</sup>	
4.7	0.9966722246	0.003322238246	5.537059587	10 <sup>-6</sup>	
4.8	0.9966722240	0.003322238892	5.537061742	10 <sup>-6</sup>	
4.9	0.9966722235	0.003322239371	5.537063338	10 <sup>-6</sup>	
5	0.9966722232	0.003322239726	5.537064520	10 <sup>-6</sup>	

**Рис. 8.12.** Определение вероятностей состояний системы массового обслуживания

В табл. 8.2 приводятся дифференциальные уравнения первого и второго порядка для самостоятельного решения аналитическими методами. Необходимо получить общее и частное решения при заданных начальных условиях. Обязательным является проверка правильности решения задачи методом подстановки. Полезным является представление решения в графической форме.

**Таблица 8.2.** Дифференциальные уравнения для самостоятельного решения

№ п/п	Уравнение	Начальные условия
1	$y' - 2,3y - x = 0$	$y(0) = 1$
2	$y' = xy$	$y(1) = 2,6$
3	$(x + y)y' + x + y - 1,5 = 0$	$y(1) = 2$
4	$y' = e^x y$	$y(0) = 0$
5	$y' = x^2 y$	$y(0) = 1$
6	$y' + 0,5x^2 y = x^2 y^2$	$y(0) = 0$
7	$y' + 2x = (x^2 + x + 2)y^2$	$y(1) = 1$
8	$y' + xy = (x + 1)y^3$	$y(1) = 1$
9	$y' = x^2 y^2$	$y(0) = 1$
10	$y' = e^{x+y}$	$y(1) = 1$
11	$(x + 1)y' + 0,5y - x = 0$	$y(1) = 1$
12	$y' = (x + 1)(y + 1)$	$y(0) = 1$
13	$y' = xe^y$	$y(0) = 2$
14	$y' + 2(x + y) = (x + 1)$	$y(0) = 0$

Таблица 8.2 (окончание)

№ п/п	Уравнение	Начальные условия
15	$y' + xy = (x+1)y^2$	$y(1) = 1$
16	$(x+y)y' + xy - 3,4 = 0$	$y(0) = 2$
17	$y'' + y' + y = x+1$	$y(0) = 1, y'(0) = 2$
18	$y'' = y+1$	$y(0) = 0, y'(0) = 1$
19	$y'' + 2y + (y')^2 = 0$	$y(1) = y'(1) = 1$
20	$y'' = e^y$	$y(0) = y'(0) = 1$
21	$y'' + 3y' + 3,4y = x^2 + x + 1$	$y(0) = y'(0) = 1$
22	$y'' + ay' + b(y')^2 = 0$	$y(0) = 0, y'(0) = 1$
23	$y'' + 3y' = x^2 - 1$	$y(0) = y'(0) = 1$
24	$y'' = \ln y$	$y(1) = y'(1) = 1$
25	$y'' + 2y' + 1,5y = x + 3$	$y(0) = y'(0) = 1,5$
26	$y'' = y + x + 1$	$y(1) = y'(1) = 5$
27	$y'' + 3,2y = x^2 + 3x + 1$	$y(1,5) = 1,6; y'(1,5) = 3,2$
28	$y'' + ay' - by = x$	$y(0) = 1,3; y'(0) = 0,5$
29	$y'' - 2y' + 3(y')^2 = 0$	$y(0) = y'(0) = 1,5$
30	$y'' + ay' + 2,5y = e^x$	$y(1) = y'(1) = 1,6$

Рассмотрим типичный пример решения дифференциального уравнения.

*Пример 8.17.* Дано дифференциальное уравнение  $y'' + 3y' + y = xe^x$ .

Необходимо:

- ☐ найти общее и частное решения при начальных условиях  $y(0) = 1, y'(0) = 1$ ;
- ☐ найти решение в явном виде;
- ☐ представить решение в виде графика;
- ☐ проверить правильность решения задачи.

*Решение задачи.*

Определение общего и частного решения.

Структуре нашего уравнения соответствует функция `DSOLVE2`.

В данном случае —  $p(x) = 3, q(x) = 1, r(x) = xe^x$ . Тогда функция `DSOLVE2` будет иметь вид:

- ☐ в случае общего решения — `DSOLVE2(3, 1, xex, x, c1, c2)`;
- ☐ в случае частного решения — `DSOLVE2_BV(3, 1, xex, x, 0, 1, 0, 1)`;  
или `DSOLVE2_IV(3, 1, xex, x, 0, 1, 1)`.

После набора и ввода функций необходимо щелкнуть мышью поочередно по кнопке **Simplify**. В результате получим:

- ☐ общее решение —  $c_1 e^{x(\frac{\sqrt{5}}{2} - \frac{3}{2})} + c_2 e^{x(\frac{\sqrt{5}}{2} + \frac{3}{2})} + e^x(\frac{x}{5} - \frac{1}{5})$ ;

- ☐ частное решение —  $\frac{6}{5} e^{x(\frac{\sqrt{5}}{2} - \frac{3}{2})} + e^x(\frac{x}{5} - \frac{1}{5})$ .

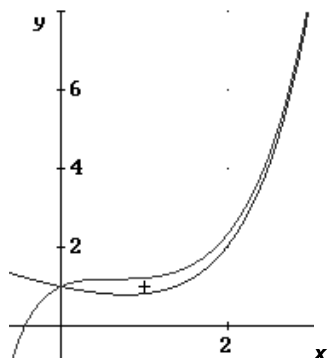
Процедуры решения показаны на рис. 8.13.

Отыскание решения в явном виде.

В нашем случае решение получено в явном виде.

Представление решения в виде графика.

Полученное решение является функцией  $y = \varphi(x)$ . Ее график получим с помощью функции `PLOT` обычным способом (рис. 8.14).

#1: DSOLVE2(3, 1, x·e<sup>x</sup>, x, c1, c2)#2: c1·e<sup>x·(√5/2 - 3/2)</sup> + c2·e<sup>-x·(√5/2 + 3/2)</sup> + e<sup>x</sup>·(x/5 - 1/5)#3: DSOLVE2\_BU(3, 1, x·e<sup>x</sup>, x, 0, 1, 0, 1)#4: 
$$\frac{6 \cdot e^{x \cdot (\sqrt{5}/2 - 3/2)}}{5} + e^x \cdot \left( \frac{x}{5} - \frac{1}{5} \right)$$
#5: DSOLVE2\_IU(3, 1, x·e<sup>x</sup>, x, 0, 1, 1)#6: e<sup>x·(√5/2 - 3/2)</sup>·(14·√5/25 + 3/5) + e<sup>-x·(√5/2 + 3/2)</sup>·(3/5 - 14·√5/25) + e<sup>x</sup>·(x/5 - 1/5)**Рис. 8.13.** Общее и частное решения дифференциального уравнения**Рис. 8.14.** График решения дифференциального уравнения

$$y'' + 3y' + y = xe^x$$

Проверка правильности решения задачи.

Проверку правильности решения выполним методом подстановки. Для этого найдем первую и вторую производные. Производные определим с помощью функции **Differentiate** (кнопка  $\partial$  на панели инструментов). В результате получим:

$$y' = e^{x(\frac{\sqrt{5}}{2} - \frac{3}{2})} \left( \frac{3\sqrt{5}}{5} - \frac{9}{5} \right) + \frac{xe^x}{5};$$

$$y'' = e^{x(\frac{\sqrt{5}}{2} - \frac{3}{2})} (\frac{21}{5} - \frac{9\sqrt{5}}{5}) + e^x (\frac{x}{5} - \frac{1}{5}).$$

С помощью команды **Sub** (на панели инструментов) подставим функции  $y(x)$ ,  $y'(x)$ ,  $y''(x)$  в исходное дифференциальное уравнение и упростим его с помощью функции **Simplify | Expan**. В результате получим ответ:  $xe^x$ , что соответствует правой части исходного дифференциального уравнения. Таким образом получено точное решение.

Решение дифференциального уравнения может быть получено в неявном виде. В этом случае его необходимо разрешить относительно искомой неизвестной. В противном случае нельзя проверить правильность решения задачи.

Рассмотренные нами функции аналитических методов решения дифференциальных уравнений позволяют получить решение уравнений первого и второго порядка. При анализе функционирования технических, информационных и других сложных систем приходится решать уравнения более высокого порядка. В этом случае пользуются численными методами. Между тем система Derive 5, как и другие универсальные математические средства символьной математики, позволяет получить аналитические решения, используя приближенные методы, рассмотренные в настоящей главе. Для этого необходимо составить программу на языке системы.

Среди приближенных численных методов наибольшее применение получил метод Рунге-Кутты. В табл. 8.3 приведены задачи, которые предлагается решить этим методом. Заданное дифференциальное уравнение, путем введения новых переменных, необходимо первоначально свести к системе дифференциальных уравнений порядка  $n$ , соответствующего порядку исходного уравнения, а затем решить эту систему, определив искомое неизвестное и все его производные. Решение можно получить в виде таблицы и графика.

**Таблица 8.3.** Задачи для самостоятельного решения  
методом Рунге-Кутты

№ п/п	Уравнение	Начальные условия				h	n
		$x_0$	$y_0$	$y'_0$	$y''_0$		
1	$y''' = x + 2y + z^2$	0	1	1	1	0,1	20
2	$y''' = x^2 + \ln(y) - z$	0	0	1	2	0,2	30
3	$y''' = xy + e^2$	1	2	1,2	2,3	0,1	20
4	$y''' = x/z + \sin(y)$	1	1,3	1	1,4	0,05	50
5	$y''' = (x+1)e^{yz}$	0	0	0	0	0,1	20
6	$y''' = xy + 0,5z$	1	1	1	1	0,1	30
7	$y''' = \cos(x) + yz$	1	1,3	1,5	1	0,2	30
8	$y''' = e^{(x+y)} - 2,5z$	1	0	1	2	0,05	20
9	$y''' = (x+y)/z$	0	1	0	1	0,1	30
10	$y''' = xyz$	0	1	1	2	0,1	20
11	$y''' = \ln(x) - z/y$	0	0	1	1	0,2	40
12	$y''' + xy'' - xz = 0$	1	2	3	1	0,1	30
13	$y''' + xyy'' - 1 = 0$	1	1	2	2	0,2	35
14	$y''' - 2xy' + z = 1$	1	1	1	1	0,1	20
15	$y''' + x + 2y'' - y' = 1$	2	2,3	2,5	2,5	0,1	30
16	$y''' + zy' - xy + 1 = 0$	2	2	3	1	0,1	36
17	$y''' + 2y' - xyz - 1 = 0$	0	0	1	1	0,1	45



Таблица 8.3 (окончание)

№ п/п	Уравнение	Начальные условия				h	n
		$x_0$	$y_0$	$y'_0$	$y''_0$		
18	$y''' + y'' - yz - 5 = 0$	0	1	1	1	0,2	20
19	$y''' - 2xz + y'' - y = 1$	0	1	2	1	0,05	25
20	$y''' + xy' - xz + 2 = 0$	0	0	1	2	0,1	16
21	$y''' + xy' + y + xz = 0$	1	1	1	2	0,1	36
22	$y''' - y'' + xyz - 1 = 0$	1	2	2,4	2,8	0,1	34
23	$y''' + y/z + z/x + 3 = 0$	0	1	0	2	0,1	40
24	$y''' + y'' + y' + y = 1$	0	1	1	2	0,1	20
25	$y''' + xy - zy' - 2 = 0$	1	1	2	2,5	0,2	34

*Пример 8.18.* Пусть необходимо решить методом Рунге-Кутты дифференциальное уравнение  $y''' + xy'' + yz - 1 = 0$  при начальных условиях:  $y(0) = y'(0) = y''(0) = 1$ . Представим уравнение в виде системы уравнений. Для этого обозначим:  $y' = x, x' = z$ . Подставляя новые переменные в исходное уравнение, получим следующую систему дифференциальных уравнений, каждое из которых первого порядка:

$$y' = x;$$

$$x' = z;$$

$$z' = -xz - yz + 1.$$

Запишем функцию RK в виде:

$$RK([x, z, -xz - yz + 1], [t, x, y, z], [0, 1, 1, 1], 0.1, 20).$$

В нашем случае будет получено решение в виде таблицы, состоящей из четырех столбцов и двадцати строк с шагом  $h = 0,1$ . Решение уравнения в диапазоне от 0 до 1 показано на рис. 8.15.

```
#1: RK([x, z, -x*z - y*z + 1], [t, x, y, z], [0, 1, 1, 1], 0.1, 20)
```

	0	1	1	1
0.1	1.105170833	1.095015791	0.9006286312	
0.2	1.221402570	1.180241391	0.8047076156	
0.3	1.349858497	1.256158104	0.7148030669	
0.4	1.491824240	1.323456987	0.6325569362	
0.5	1.648720638	1.382951608	0.5588006773	
0.6	1.822117962	1.435504565	0.4937108627	
0.7	2.013751626	1.481970368	0.4369766429	
0.8	2.225539563	1.523154750	0.3879582440	
0.9	2.459601413	1.559788849	0.3458243387	
#2:	1	2.718279744	1.592515820	0.3096628468

**Рис. 8.15.** Решение системы дифференциальных уравнений методом Рунге-Кутты

В системе Derive 5 имеется функция, выделяющая из матрицы  $A$  столбцы  $j$  и  $k$ . Она может быть использована для графического изображения решения дифференциального уравнения. Функция имеет следующий вид:

```
EXTRACT_2_COLUMNS(A, j, k).
```

Воспользуемся ею и представим наше решение в виде графика. Технология построения графика функции  $y = f(t)$  относительно проста и включает ряд действий.

- Образование матрицы, состоящей из двух столбцов  $t$  и  $y$ , которые в матрице решения нашего дифференциального уравнения находятся в первом и третьем столбцах.

Пусть матрица решения уравнения находится в четвертой строке экрана (#4). Тогда функция выделения столбцов матрицы будет иметь вид:

```
EXTRACT_1_COLUMNS(#4, 1, 3).
```

После ввода этой функции и нажатия кнопки **Approximate** (расположена на панели инструментов) на экране образуется следующая матрица:

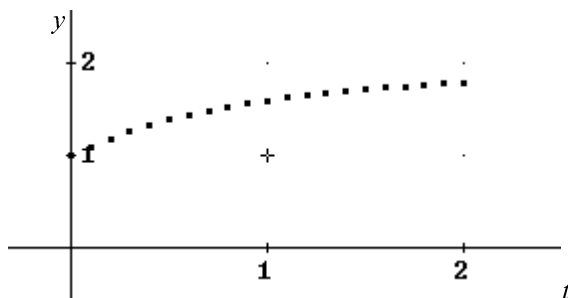
0	1
0.1	1.095015790

0.2	1.180241391
...	
0.9	1.559788848
1	1.592515820

- Щелчок по кнопке **2D-plot window** (расположена на панели инструментов).

В результате на экране формируется графическое окно **2D-plot** с изображением координатной сетки.

- Щелчок мышью по кнопке **Plot Expression**. На экране формируется график функции  $y = f(t)$ , являющейся решением дифференциального уравнения (рис. 8.16).



**Рис. 8.16.** Решение дифференциального уравнения  $y''' + xy'' + yz - 1 = 0$

## Глава 9



# Вычисление интегралов

Система Derive 5 имеет богатые возможности символьных вычислений. Она позволяет вычислять неопределенные интегралы, подынтегральная функция которых задана в виде аналитических выражений. Это дает возможность вычислить интеграл при любых значениях переменных, пользуясь методом Ньютона:

$$\int_a^b f(x)dx = F(x)\Big|_a^b = F(b) - F(a),$$

где  $F(x)$  — аналитическое выражение интеграла.

Общность решения — основное достоинство аналитических методов.

Если неопределенный интеграл вычислить невозможно, то обращаются к численным методам интегрирования. Численное интегрирование необходимо в следующих случаях:

- ☐ первообразная не выражается через элементарные функции. Здесь "первообразная" оставить;
- ☐ аналитическое выражение интеграла слишком сложно;
- ☐ подынтегральная функция задана в табличной форме.

При вычислениях интегралов символьными и численными методами подынтегральную функцию целесообразно представлять в наиболее простом виде. Это может существенно ускорить вычисления (особенно в аналитическом виде). Упрощение подынтегральной функции можно выполнить, воспользовавшись функцией

**Simplify**, вызываемой из главного меню системы. Имеют место случаи (и не редкие), когда система до упрощения не может вычислить неопределенный интеграл и легко его определяет после упрощения.

Вычисление интеграла в Derive 5 осуществляется с помощью методов, которые пользователю неизвестны.

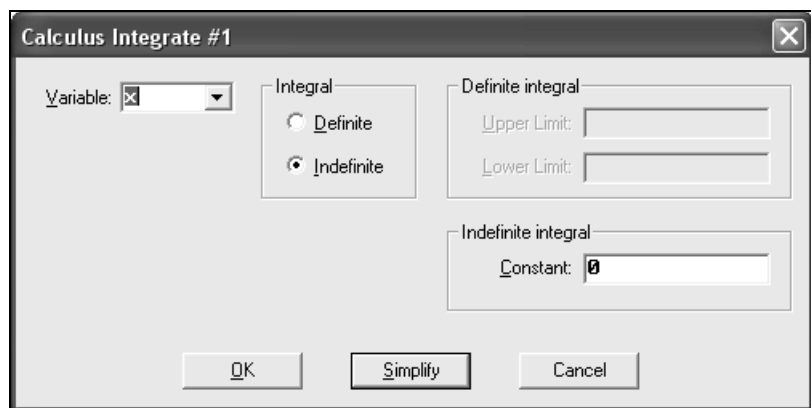
Пользователь вводит подынтегральную функцию, указывает переменную интегрирования, вид интегрирования (определенный или неопределенный интеграл) и пределы  $a, b$ . В результате получает ответ. При этом неизвестен ни метод интегрирования, ни его точность. Пользователь может лишь установить число знаков ответа, которое, к сожалению, не является критерием точности.

## 9.1. Технология вычисления интегралов в системе Derive 5

Вычисление интегралов в системе Derive 5 осуществляется с помощью функции **Integrate**. Технология вычисления интеграла состоит в следующем:

- ввод подынтегральной функции  $f(x)$ ;
- выполнение команды **Calculus | Integrate** или щелчок по кнопке  $\int$ , расположенной на панели инструментов (на экране формируется окно **Calculus Integrate #1**, где #1 — номер строки подынтегральной функции). В данном случае предполагается, что подынтегральная функция находится в первой строке экрана (рис. 9.1);
- установка в области **Variable** переменной интегрирования;
- щелчок мышью в области панелей **Indefinite** (при вычислении неопределенного интеграла) или **Definite** (при вычислении определенного интеграла) с последующей установкой значений — **Upper Limit**, **Lower Limit** для **Definite integral** (соответственно верхнего и нижнего пределов интегрирования; в области **Indefinite integral**; записывается постоянная интегрирования;

- ☐ щелчок мышью по кнопке **Simplify** (на экране формируется значение интеграла).



**Рис. 9.1.** Окно установки параметров интегрирования

$$\int \frac{x}{1 - x^2} dx$$

$$- \frac{\ln(x^2 - 1)}{2}$$

$$\int_0^2 \frac{x^2}{\sqrt{1 + x^2}} dx$$

**Рис. 9.2.** Вычисление интегралов неопределенного и определенного

1.514250239

$$\frac{\text{SIN}(x)}{(1+x)!}$$

$$\int \frac{\text{SIN}(x)}{(1+x)!} dx$$

$$\int \frac{\text{SIN}(x)}{(x+1)!} dx$$

Рис. 9.3. Вычисление интеграла неберущегося

Ниже показаны результаты вычисления трех интегралов: неопределенного; определенного; неберущегося.

Ответом является (рис. 9.2, 9.3): в первом случае выражение, во втором — число, в третьем повторяется выражение интеграла, что означает отсутствие решения.

## 9.2. Алгоритмы численных методов вычисления интегралов

Существует ряд способов численного интегрирования. Во всех этих способах вычисление осуществляется по приближенным формулам, называемым квадратурными. Рассмотрим некоторые из них.

### 9.2.1. Формулы прямоугольников

Формулы прямоугольников имеют вид:

$$\int_a^b f(x) dx = \left\{ \begin{array}{l} h \sum_{k=0}^{n-1} y_k \\ h \sum_{k=1}^n y_k \end{array} \right\}, \quad (9.1)$$

где  $h$  — шаг интегрирования;  $y_k$  — значение подынтегральной функции при аргументе  $x_k, k = 0, 1, 2, \dots, n$ ;  $n = \frac{b-a}{h}$  — число частей, на которые разбивается область интегрирования  $(a-b)$ . Одна из формул дает значение интеграла с избытком, другая с недостатком. Какая из них выдает решение с избытком или с недостатком, зависит от вида подынтегральной функции.

### 9.2.2. Формула трапеций

Данная формула имеет вид:

$$\int_a^b f(x)dx = h \left( \frac{y_0}{2} + \sum_{k=1}^{n-1} y_k + \frac{y_n}{2} \right), \quad (9.2)$$

где  $y_0$  — значение подынтегральной функции при  $x = a$ ;  $y_n$  — значение подынтегральной функции при  $x = b$ .

### 9.2.3. Формула парабол (Симпсона)

Формула парабол имеет вид:

$$\begin{aligned} \int_a^b f(x)dx = & \quad (9.3) \\ = \frac{h}{3} (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 + \dots + 4y_{n-1} + y_n). \end{aligned}$$

В ней ординаты с нечетными индексами умножаются на 4, а с четными на 2. Предполагается, что  $n$  — число четное. При нечетном  $n$  формула имеет следующий вид:

$$\int_a^b f(x)dx = \frac{h}{3} (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + 4y_5 + \dots + 2y_{n-1} + y_n).$$

Здесь крайние ординаты имеют коэффициент, равный 1.



Существует много других квадратурных формул вычисления интегралов (Котеса, Чебышева, Гаусса и др.). Вычисление интеграла с помощью квадратурных формул требует знания числа  $n$  области интегрирования  $(a - b)$ . Значение  $n$  выбирается из условий требуемой точности вычисления интеграла. Зависимости погрешности —  $\varepsilon$  от  $n$ , в общем случае, не существует. В связи с этим, для обеспечения точности вычисления интегралов используют следующий алгоритм.

Определенный интеграл вычисляется дважды, с произвольным шагом  $h$  и  $\frac{h}{2}$ . Если при этом разность интегралов мала, то вычислительный процесс заканчивается, а значение интеграла считается то, которое вычислено с шагом  $\frac{h}{2}$ . В противном случае шаг уменьшется вдвое и сравниваются значения интеграла с шагом  $\frac{h}{2}$  и  $\frac{h}{4}$  и т. д.

Вычисление интеграла численными методами с помощью Derive 5 может оказать большую помощь учащемуся при изучении численных методов и программирования. С помощью Derive 5 пользователь может анализировать подынтегральную функцию, что необходимо в случаях, когда разработанная программа не дает возможности вычислить интеграл с необходимой точностью (например, при наличии разрывов непрерывности подынтегральной функции). Сравнение результатов, полученных с помощью программы и посредством Derive 5, позволяет убедиться в достоверности решения и отсутствии ошибок в программе учащегося.

### 9.3. Вычисление определенных интегралов по квадратурным формулам

Derive 5 позволяет легко реализовать квадратурные формулы прямоугольников, трапеций, Симпсона и др.

Вычисление интеграла по формулам прямоугольников требует вычисления значений суммы ординат подынтегральной функции. Вычислить ординаты функции  $f(x)$  можно путем ее табулирования, а значение суммы полученного вектора можно получить с помощью команды **Sum**.

Алгоритмом вычисления интеграла по методу прямоугольников является выражение (9.1). Составим программу реализации этого алгоритма на языке системы Derive 5. Схема алгоритма будет линейной и состоящей из следующих четырех блоков.

- Блок 1 — табулирование подынтегральной функции с шагом  $h$  в диапазоне изменения аргумента, равного пределам интегрирования  $(a - b)$ .
- Блок 2 — выбор диапазона изменения значений вектора подынтегральной функции, полученного в результате ее табулирования.
- Блок 3 — вычисление суммы значений подынтегральной функции в диапазоне пределов интегрирования с шагом  $h$ .
- Блок 4 — вычисление значения интеграла по формуле прямоугольников.

Создадим программу вычисления интеграла, соответствующую данной блок-схеме. Для этого выполним следующие действия.

1. Набор и ввод функции табуляции `VECTOR(f, x, a, b, h)`.

В результате на экране, в строке #1, появится выражение:

```
#1: VECTOR(f, x, a, b, h),
```

где  $f$  — подынтегральная функция;  $x$  — аргумент подынтегральной функции;  $a, b$  — пределы интегрирования;  $h$  — шаг интегрирования.

2. Набор и ввод функции `ELEMENT(#1, n)`. Причем:  $\#1$  — номер строки, в которой находится функция табуляции;  $n$  — символ, означающий диапазон элементов суммирования.

На экране в строке #2 появится выражение:

```
#2: ELEMENT(VECTOR(f, x, a, b, h), n).
```

Образование суммы элементов вектора значений подынтегральной функции. Для этого:

- выделив строку #2, щелкнуть мышью по пункту главного меню **Calculus** | **Sum** или по кнопке  $\Sigma$ , расположенной на панели инструментов (на экране появится диалоговое окно **Calculus Sum**, показанное на рис. 9.1, с тремя вкладками;
- в области **Variable** следует установить переменную суммирования  $n$ , установить флажок **Definite** в области **Sum** и ввести пределы суммирования в строках ввода панели **Definite integral**: нижний (**Lower Limit**) и верхний (**Upper Limit**) (пусть это будут соответственно —  $nL$  и  $nU$ ).

После щелчка по кнопке **OK** на экране появится выражение:

$$\#3: \sum_{n=nL}^{nU} \text{ELEMENT}(\text{VECTOR}(f, x, a, b, h), n).$$

Образование конечной формулы вычисления интеграла. Для этого необходимо набрать и ввести выражение —  $h * \#3$ .

На экране появится следующий результат:

$$\#4: h \sum_{n=nL}^{nU} \text{ELEMENT}(\text{VECTOR}(f, x, a, b, h), n).$$

Программа вычисления интеграла по методу прямоугольников создана. Она является универсальной, позволяющей получить значение интеграла любой функции  $f(x)$  с любыми пределами интегрирования. Программа имеет вид выражения #4.

Вычислим интеграл по созданной программе.

*Пример 9.1.* Пусть необходимо вычислить  $\int_a^b \frac{x}{1+x^2} dx$  методами

прямоугольников и трапеций. При этом значение интеграла, вычисленного по методу трапеций, равно полусумме интегралов, вычисленных по методу прямоугольников с достоинством и недостатком. Интеграл вычислим в диапазоне интегрирования от 0 до 5 и шагом  $h = 0,1$ ;  $h = 0,01$ .

В данном случае число участков, на которые разбивается весь диапазон интегрирования, будет: при  $h = 0,1$  —  $n = \frac{b-a}{h} = \frac{5-0}{0,1} = 50$ ; при  $h = 0,01$  —  $n = \frac{b-a}{h} = \frac{5-0}{0,01} = 500$ .

Тогда сумму значений подынтегральной функции при  $h = 0,1$  необходимо вычислять в диапазоне  $nL=1$  и  $nU=49$ , при вычислении интеграла с недостатком, и в диапазоне  $nL=2$  и  $nU=50$ , при вычислении с избытком. При шаге  $h = 0,01$  диапазон изменения суммы значений подынтегральной функции будет:  $nL=1$  и  $nU=499$ , при вычислении интеграла с недостатком;  $nL=2$  и  $nU=500$ , при вычислении интеграла с избытком.

Технология решения задачи.

- Выделим строку #4 программы.
- Щелкнем мышью по кнопке **Sub**. На экране появится окно **Substitute for Variables**, имеющее две вкладки. Во вкладке **Variables** перечислены имена переменных  $a, b, f, h, nL, nU$ , а вкладка **New Value** пустая с мигающим курсором.
- Поочередно щелкая мышью по переменным и пустой вкладке, вводим переменные:  $a=0$ ,  $b=5$ ,  $f=x/(1+x^2)$ ,  $y=0.1$ ,  $nL=1$ ,  $nU=49$ . После окончания ввода щелкаем мышью по кнопке **OK**.

В результате проделанных действий на экране появляется выражение:

$$\#5: 0.1 \sum_{n=1}^{49} \text{ELEMENT} \left( \text{VECTOR} \left( \frac{x}{1+x^2}, x, 0, 5, 0.1 \right), n \right)$$

- Для получения ответа необходимо щелкнуть мышью по кнопке **Approx**, расположенной на панели инструментов.

На экране формируется значение интеграла с недостатком:

$$\#6: 1.598976965.$$

- Для получения интеграла с избытком отредактируем выражение в строке #5. Для этого выделим его и нажмем клавишу <F3> (в строке пользователя появится выражение, находящееся

в строке #5). Отредактируем числа 1 и 49, заменив их числами 2 и 50 и введем выражение, нажав клавишу <Enter>.

На экране появится новое выражение:

$$\#7: 0.1 \sum \text{ELEMENT} \left( \text{VECTOR} \left( \frac{x}{1+x^2}, x, 0, 5, 0.1 \right), n \right).$$

□ Щелкнуть мышью по кнопке **Approx** ( $\approx$ ), расположенной на панели инструментов.

В результате последнего действия на экране будет сформировано значение интеграла с избытком:

$$\#8: 1.618569128.$$

Для получения решения по методу трапеций достаточно найти среднее значение интеграла, полученного по методу прямоугольников с недостатком и с избытком. Для этого достаточно ввести выражение  $(\#6 + \#8) / 2$  и щелкнуть мышью по кнопке **Approx**. На экране появится выражение среднего и значение интеграла, вычисленного по методу трапеций:

$$\#9: \frac{1.598976965 + 1.6185699128}{2}$$

$$\#10: 1.608773046$$

Вычислим теперь значение интеграла по методу прямоугольников с недостатком и избытком при  $h = 0,01$ . Для этого достаточно отредактировать выражение #7, изменяя значения  $h$  и  $n$ . В результате будут получены следующие значения интеграла: с недостатком — 1.626151467, с избытком — 1.628078101. Тогда значение интеграла, вычисленное по методу трапеций, будет определяться как 1.627114784.

Оценим точность полученных решений. Для этого вычислим интеграл с помощью функции **Integrate** по методике, которая описана в *разд. 9.1* настоящей главы. Получим точное значение интеграла в виде:

$$\frac{LN(26)}{2} = 1,629048269.$$

Сравнивая результаты расчетов, видим, что значение интеграла, вычисленное по методу прямоугольников и трапеций с шагом 0,1, верно с точностью

лишь один знак после запятой, а при шаге 0,01 точность решения повысилась на порядок. Теперь решение удовлетворяет точности два знака после запятой.

Программа и решение задачи на экране монитора имеют следующий вид:

```
#1: VECTOR(f, x, a, b, h)
#2: ELEMENT(VECTOR(f, x, a, b, h), n)
#3:  $\sum_{n=nn}^{nr} \text{ELEMENT}(\text{VECTOR}(f, x, a, b, h), n)$ 
#4:  $h \sum_{n=nn}^{nr} \text{ELEMENT}(\text{VECTOR}(f, x, a, b, h), n)$ 
#5:  $0.1 \sum_{n=1}^{49} \text{ELEMENT}(\text{VECTOR}(\frac{x}{1+x^2}, x, 0, 5, 0.1), n)$ 
#6: 1.598976965
#7:  $0.1 \sum_{n=1}^{50} \text{ELEMENT}(\text{VECTOR}(\frac{x}{1+x^2}, x, 0, 5, 0.1), n)$ 
#8: 1.618569128
#9:  $\frac{1.598976965 + 1.618569128}{2}$ 
#10: 1.608773046
#11:  $0.01 \sum_{n=1}^{499} \text{ELEMENT}(\text{VECTOR}(\frac{x}{1+x^2}, x, 0, 5, 0.01), n)$ 
#12: 1.626151467
#13:  $0.01 \sum_{n=2}^{500} \text{ELEMENT}(\text{VECTOR}(\frac{x}{1+x^2}, x, 0, 5, 0.01), n)$ 
#14: 1.628078101
#15:  $\frac{1.626151467 + 1.628078101}{2}$ 
#16: 1.627114784
```

### 9.3.1. Вычисление интеграла по методу парабол (Симпсона)

Первоначально разработаем программу вычисления интеграла на языке системы Derive 5. В соответствии с формулой 9.3, схема алгоритма будет состоять из следующих блоков.

- Блок вычисления суммы значений подынтегральной функции с нечетными индексами.
- Блок вычисления суммы значений подынтегральной функции с четными индексами.
- Блок вычисления интеграла.

Аналогично программе вычисления интеграла по методу прямоугольников блок вычисления значений суммы реализуется тремя функциями: VECTOR, ELEMENT, SUM. Функция VECTOR табулирует подынтегральную функцию в диапазоне изменения аргумента от  $(a+h)$  до  $(b-h)$  с шагом  $2h$  (т. е. в диапазоне, соответствующем значениям функции с нечетными индексами). После набора и ввода функции на экране она будет иметь вид:

```
#1: VECTOR(f, a+h, b-h, 2h).
```

Для определения суммы значений функции необходимо указать переменную суммирования. Для этого используется функция ELEMENT( $\#1, n1$ ), где  $\#1$  — номер строки функции VECTOR,  $n1$  — переменная суммирования. После набора и ввода этой функции на экране появится выражение:

```
#2: ELEMENT(VECTOR(f, a+h, b-h, 2h), n1).
```

Определим диапазон изменения переменной суммирования  $n1$ .

Число участков в диапазоне интегрирования  $n = \frac{b-a}{h}$ . Тогда

общее число значений функции будет равно  $\frac{b-a}{h} + 1$ . Из этого

числа ординат необходимо вычесть две —  $y_0$  и  $y_n$ . В результате получим общее число функций с четными и нечетными индексами. Будем считать, что половина из них — это функции с четными (нечетными) индексами. Тогда общее их число бу-

дет:  $\frac{\frac{b-a}{h} + 1 - 2}{2} = \frac{b-a-h}{2h}$ . Из этого следует, что переменная суммирования  $n1$  должна изменяться в пределах от 1 до  $\frac{b-a-h}{2h}$ . Выделив строку #2 и щелкнув мышью по кнопке  $\Sigma$ , расположенной на панели инструментов, вызовем окно **Calculus Sum** с тремя панелями. Введем переменную суммирования  $n1$  в области **Variable** и пределы суммирования в полях ввода панели **Definite Sum**. Теперь осталось щелкнуть мышью по кнопке **OK** и на экране появится выражение:

$$\#3: \sum_{n1=1}^{(b-a-h)/(2h)} \text{ELEMENT}(\text{VECTOR}(f, x, a+h, b-h, 2h), n1).$$

Сумма ординат подынтегральной функции с нечетными индексами образуется аналогично. Отличие лишь в диапазоне табулирования подынтегральной функции, который будет иметь начальное значение  $a+2h$ , конечное  $b-2h$ . Функции на экране имеют следующий вид:

$$\#4: \text{VECTOR}(f, a+2h, b-2h, 2h)$$

$$\#5: \text{ELEMENT}(\text{VECTOR}(f, x, a+2h, b-2h, 2h), n2)$$

$$\#6: \sum_{n2=1}^{(b-a-h)/(2h)} \text{ELEMENT}(\text{VECTOR}(f, x, a+2h, b-2h, 2h), n2).$$

Блок вычисления интеграла реализует конечную формулу метода парабол:  $S=h/3(fz+4\#3+2\#6+fw)$ , где  $fz$  и  $fw$  — значения подынтегральной функции при  $x=a$  и  $x=b$ . Символы  $\#3$  и  $\#6$  обозначают суммы ординат с четными и нечетными индексами, находящимися в третьей и шестой экранных строках. После набора и ввода выражения  $S$  на экране будет отображена формула парабол на языке системы Derive 5:

$$\#7: S = \frac{h}{3} \left( fz + 4 \sum_{n1=1}^{(b-a-h)/(2h)} \text{ELEMENT}(\text{VECTOR}(f, x, a+h, b-h, 2h), n1) + \right. \\ \left. + 2 \sum \text{ELEMENT}(\text{VECTOR}(f, x, a+2h, b-2h, 2h), n2) + fw \right)$$



*Пример 9.2.* Пусть требуется вычислить интеграл  $S = \int_0^5 \frac{x}{1+x^2} dx$

по методу парабол с шагом  $h = 0,1$  и  $h = 0,01$  и сравнить с результатами, полученными по методу прямоугольников, и точным методом.

Введем исходные данные, воспользовавшись кнопкой **Sub**, расположенной на панели инструментов. В нашем случае

$$a = 0, b = 5, h = 0,1, f = \frac{x}{1+x^2}, fz = \frac{z}{1+z^2}, fw = \frac{w}{1+w^2}.$$

После ввода данных необходимо щелкнуть мышью по кнопке **OK**. В результате на экране появится выражение интеграла, в котором неизвестными будут переменные  $z$  и  $w$ . В нашем случае  $z = 0$ ,  $w = 5$ . Введем эти данные с помощью кнопки **Sub** и вновь щелкнем мышью по кнопке **OK**. На экране будет сформировано окончательное выражение для вычисления интеграла. Для получения ответа щелкнем мышью по кнопке **Approx**, расположенной на панели инструментов. В результате на экране появится следующий ответ:  $S = 1.602928797$ . Отредактируем конечное выражение, заменив шаг интегрирования с  $h = 0,1$  на  $h = 0,01$ . Результатом решения будет —  $S = 1.626479424$ . Программа вычисления интеграла по методу парабол и процедуры решения задачи на экране монитора будут иметь следующий вид:

#1:  $VECTOR(f, x, a + h, b - h, 2h)$

#2:  $ELEMENT(VECTOR(f, x, a + h, b - h, 2h), n1)$

#3:  $VECTOR(f, x, a + 2h, b - 2h, 2h)$

#4:  $ELEMENT(VECTOR(f, x, a + 2h, b - 2h, 2h), n2)$

#5:  $\sum_{n1=1}^{(b-a-h)/(2h)} ELEMENT(VECTOR(f, x, a + h, b - h, 2h), n1)$

#6:  $\sum_{n2=1}^{(b-a-h)/(2h)} ELEMENT(VECTOR(f, x, a + 2h, b - 2h, 2h), n2)$

$$\begin{aligned} \#7: S = & \frac{h}{3} (fz + 4 \sum_{n1=1}^{(b-a-h)/(2h)} ELEMENT(VECTOR(f, x, a+h, b-h, 2h), n1) + \\ & + 2 \sum_{n2=1}^{(b-a-h)/(2h)} ELEMENT(VECTOR(f, x, a+2h, b-2h, 2h), n2) + fw) \end{aligned}$$

$$\begin{aligned} \#8: S = & \frac{0.1}{3} \left( \frac{z}{1+z^2} + \right. \\ & + 4 \sum_{n1=1}^{(5-0-0.1)/(2*0.1)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0+0.1, 5-0.1, 2*0.1), n1) + \\ & + 2 \sum_{n2=1}^{(5-0-0.1)/(2*0.1)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0+2*0.1, 5- \\ & \left. - 2*0.01, 2*0.1), n2) + \frac{w}{1+w^2} \right) \end{aligned}$$

$$\begin{aligned} \#9: S = & \frac{0.1}{3} \left( \frac{5}{1+5^2} + 4 \sum_{n1=1}^{(5-0-0.1)/(2*0.1)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0+ \right. \\ & + 0.1, 5-0.1, 2*0.1), n1) + \\ & + 2 \sum_{n2=1}^{(5-0-0.1)/(2*0.1)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0+2*0.1, 5- \\ & \left. - 2*0.01, 2*0.1), n2) + \frac{0}{1+0^2} \right) \end{aligned}$$

$$\#10: \quad S=1.602928797$$

$$\begin{aligned} \#11: S = & \frac{0.01}{3} \left( \frac{5}{1+5^2} + 4 \sum_{n1=1}^{(5-0-0.01)/(2*0.1)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0+ \right. \\ & + 0.01, 5-0.01, 2*0.01), n1) + \end{aligned}$$

$$+ 2 \sum_{n2=1}^{(5-0-0.01)/(2*0.01)} ELEMENT(VECTOR(\frac{x}{1+x^2}, x, 0 + \\ + 2*0.01, 5-2*0.01, 2*0.01), n2) + \frac{0}{1+0^2})$$

# 12:

S=1.626479424

Из сравнения результатов расчетов видно, что при шаге  $h = 0,01$  метод парабол обеспечивает точность вычисления интеграла два знака после запятой. Значение интеграла практически то же, что и полученное по методу трапеций. Следует при этом иметь в виду, что наш вывод не является общим для рассмотренных методов. Это лишь результат вычисления конкретного интеграла.

При вычислении несобственных интегралов с бесконечными пределами следует указывать тип переменной (положительная, отрицательная и т. п.). Пусть, например, необходимо вычислить интеграл:

$$\int_0^{\infty} e^{-ax} dx.$$

Если значение переменной  $a$  отрицательное, то интеграл будет расходящимся, и Derive 5, после нажатия кнопки **Simplify**, лишь повторит выражение интеграла (возможно в преобразованном виде). Системе необходимо указать, что во всей положительной области значений  $t$  переменная  $a$  является положительной. Для этого необходимо выполнить команды: **Declare | Variable Domain** (откроется окно, показанное на рис. 9.4).

В строке ввода **Variable Name** необходимо ввести переменную  $a$ . Окно активизируется, и в области **Interval** установить флажок **Positive(0, ∞)**. После щелчка мыши по кнопке **OK** на экране появится сообщение:  $a: \text{IReal}(0, \infty)$ .

Теперь вычислим интеграл указанным выше способом. Ответом будет  $1/a$ .

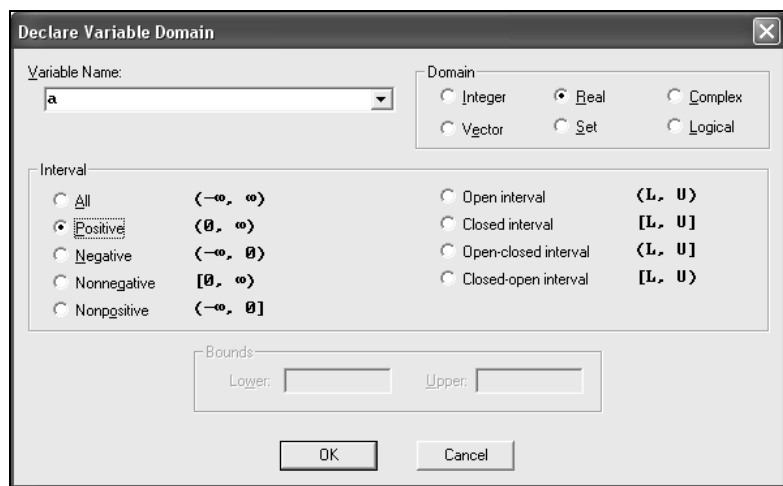


Рис. 9.4. Окно установки диапазона изменения переменной

## 9.4. Вычисление кратных интегралов

N-кратный интеграл можно вычислить, повторяя вычисление интеграла N раз (каждый раз от предыдущего результата). Такой способ показан на рис. 9.5, где получен тройной интеграл от выражения

$\frac{x-1}{x+1}$  путем его трехкратного интегрирования:

$$\begin{aligned}
 & \frac{x-1}{x+1} \\
 & \int \frac{x-1}{x+1} dx \\
 & \quad x - 2 \cdot \text{LN}(x+1) \\
 & \int \int \frac{x-1}{x+1} dx dx \\
 & \quad - 2 \cdot (x+1) \cdot \text{LN}(x+1) + \frac{x^2}{2} + 2 \cdot x \\
 & \int \int \int \frac{x-1}{x+1} dx dx dx \\
 & \quad \frac{x \cdot (x^2 + 9 \cdot x + 6)}{6} - (x^2 + 2 \cdot x + 1) \cdot \text{LN}(x+1)
 \end{aligned}$$

Рис. 9.5. Вычисление трехкратного интеграла

Derive 5 позволяет вычислить многократный интеграл без его промежуточных значений. В этом случае используется следующая технология.

□ Набор и ввод подынтегрального выражения.

□ Представление на экране общего вида интеграла:

- щелчок мыши по кнопке **Integrate** на панели инструментов (на экране появится окно **Calculus Integrate**);
- в области **Variable** ввод переменной интегрирования —  $x$ ;
- в области **Integral** активизация кнопки **Indefinite** (неопределенный интеграл);
- щелчок по кнопке **OK** (на экране формируется выражение интеграла  $! \frac{x-1}{x+1} dx$ ).

□ Представление многократного интеграла. Для этого:

- выделение выражения  $! \frac{x-1}{x+1} dx$ ;
- щелчок мышью по кнопке **!**, расположенной на панели инструментов, и в появившемся окне **Calculus Integrate** щелчок по кнопке **OK** (на экране появится двойной интеграл —  $!! \frac{x-1}{x+1} dx dx$ );
- повторяя эти процедуры  $n$  раз, получим выражение  $N$ -кратного интеграла.

□ Выделение конечного выражения  $N$ -кратного интеграла и щелчок мышью по кнопке **Simplify** ( $\approx$ ) (формируется ответ в виде математического выражения).

На рис. 9.5 показано вычисление трехкратного интеграла по приведенной технологии.

При вычислении  $N$ -кратного определенного интеграла необходимо в области **Integral** (окно **Calculus Integrate**) щелкнуть по пункту **Definite** (определенный интеграл) и затем по кнопке **Simplify**.

Например, получим значение N-кратного интеграла в пределах от  $a$  до  $b$ . Ниже показано вычисление четырехкратного интеграла, причем промежуточным является результат вычисления трехкратного интеграла в пределах от 0 до 10 (рис. 9.6).

$$\begin{aligned}
 & \frac{x-1}{x+1} \\
 & \int \frac{x-1}{x+1} dx \\
 & \int \int \frac{x-1}{x+1} dx dx \\
 & \int_0^{10} \int \int \frac{x-1}{x+1} dx dx dx \\
 & \int \int_0^{10} \int \int \frac{x-1}{x+1} dx dx dx dx \\
 & \frac{980 \cdot x}{3} - 121 \cdot x \cdot \text{LN}(11) \\
 & 36.52133865 \cdot x
 \end{aligned}$$

**Рис. 9.6.** Вычисление четырехкратного определенного интеграла

## 9.5. Примеры вычисления интегралов

На табл. 9.1—9.3 приводятся задачи на интегрирование с помощью системы Derive 5. Эти задачи будут способствовать изучению компьютерных технологий вычисления интегралов. Кроме того, они весьма оригинальны и вызывают восхищение своими ответами. Не следует огорчаться, если система выдает иное решение, чем в ответе. Вероятнее всего они идентичны. Попробуйте преобразовать полученный ответ с помощью функций меню **Simplify**.

Возможны случаи, когда система вообще не выдает решений, а оно имеется. Это доказывает лишь то, что компьютер не может

быть более интеллектуален, чем человек, т. е. его программа не совершенна.

**Таблица 9.1.** Таблица неопределенных интегралов

№ п/п	Интеграл	Ответ
1	$\int \frac{x}{1+x^2} dx$	$\frac{1}{2} \ln(1+x^2)$
2	$\int \frac{x}{1+x^4} dx$	$\frac{1}{2} \operatorname{arctg} x^2$
3	$\int \frac{1}{1+x^2} dx$	$a \tan(x)$
4	$\int \frac{x}{1-x^4} dx$	$-\frac{1}{4} \ln \frac{-1+x^2}{1+x^2}$
5	$\int \frac{dx}{x(1+x^2)}$	$\ln \frac{x}{\sqrt{1+x^2}}$
6	$\int \frac{dx}{\sqrt{(a+bx)^3}}$	$-\frac{2}{b\sqrt{a+bx}}$
7	$\int \frac{xdx}{\sqrt{(a+bx+cx^2)^3}}$	$-\frac{2(2a+bx)}{(4ac-b^2) \cdot \sqrt{a+bx+cx^2}}$
8	$\int \frac{x^2 dx}{\sqrt{(a+bx)^3}}$	$\frac{2(b^2 x^2 - 4abx - 8a^2)}{3b^2 \sqrt{a+bx}}$
9	$\int \frac{x^4}{(\sqrt{a+cx^2})^7} dx$	$\frac{1}{5} - \frac{x^5}{a(\sqrt{a+cx^2})^5}$

Таблица 9.1 (продолжение)

№ п/п	Интеграл	Ответ
10	$\int \frac{dx}{x\sqrt{a+cx^2}}$	$\frac{1}{\sqrt{a}} \ln \frac{\sqrt{a+cx^2} - \sqrt{a}}{x}$
11	$\int \frac{dx}{\sinh(x) \cdot \cosh(x)}$	$\ln \tanh(x)$
12	$\int \frac{dx}{\sinh^2 x \cdot \cosh^2 x}$	$\frac{4}{1-e^{4x}}$
13	$\int \frac{ch(x)dx}{ch(x)+sh(x)}$	$\frac{x}{2} - \frac{1}{4}e^{-2x}$
14	$\int xe^{ax} \sinh(ax)dx$	$\frac{e^{2ax}}{4a} \left( x - \frac{1}{2a} \right) - \frac{x^2}{4} + \frac{1}{8a^2}$
15	$\int \sin(x) \cos^2(x)$	$-\frac{\cos^3(x)}{3}$
16	$\int \sin^4(x) \cos(x)dx$	$\frac{\sin^5(x)}{5}$
17	$\int \frac{1}{\sin(x)} dx$	$\ln(\tan(\frac{x}{2}))$
18	$\int \frac{x + \sin x}{1 + \cos x} dx$	$x \operatorname{tg} \frac{x}{2}$
19	$\int \frac{1}{x^2(x-1)} dx$	$\ln(x-1) - \ln(x) + \frac{1}{x}$
20	$\int \frac{x \sin x}{\cos^2 x} dx$	$\frac{x}{\cos x} - \ln \operatorname{tg} \left( \frac{x}{2} + \frac{\pi}{4} \right)$



**Таблица 9.1** (окончание)

№ п/п	Интеграл	Ответ
21	$\int \frac{dx}{\sin^3 x \cdot \cos x}$	$-\frac{1}{2 \sin^2 x} + \ln \operatorname{tg} x$
22	$\int x \cos(x^2) dx$	$\frac{\sin x^2}{2}$
23	$\int \frac{\sin^2(x) dx}{\cos^4(x)}$	$\frac{1}{3} \operatorname{tg}^3(x)$
24	$\int a^x e^{-x} dx$	$\frac{e^{-x} a^x}{\ln a - 1} - \frac{1}{\ln a - 1}$

**Таблица 9.2.** Таблица определенных интегралов

№ п/п	Интеграл	Ответ
1	$\int_0^1 \frac{x dx}{\sqrt{1-x}}$	$\frac{4}{3}$
2	$\int_0^1 \frac{1}{(1-ax)^n} dx$	$\frac{(1-a)^{1-n} - 1}{a(n-1)}$
3	$\int_0^1 x^{n-1} (x+1) dx$	$\frac{1}{n+1} + \frac{1}{n}$
4	$\int_0^1 \left( \frac{x^2}{x+1} \right) dx$	$\ln 2 - \frac{1}{2}$
5	$\int_0^1 \frac{x^5}{1+x} dx$	$\frac{47}{60} - \ln 2$

Таблица 9.2 (окончание)

№ п/п	Интеграл	Ответ
6	$\int_0^1 \frac{x^2 dx}{\sqrt{1-x^2}}$	$\frac{\pi}{4}$
7	$\int_0^1 x \sin x dx$	$\sin(1) - \cos(1) = 0,301$
8	$\int_0^1 x \cos x dx$	$\cos(1) - \sin(1) - 1$
9	$\int_0^1 x^n \ln x dx$	$-\frac{1}{(n+1)^2}$
10	$\int_0^1 \frac{x^3 \ln x}{1+x} dx$	$-0,0386$

Таблица 9.3. Таблица несобственных интегралов

№ п/п	Интеграл	Ответ
1	$\int_0^{\infty} \frac{1+x}{(x+a)^{p+1}} dx$	$(x+1)(x+a)^{-p-1}$
2	$\int_0^{\infty} x^a e^{-x} dx$	$a!$
3	$\int_{-1}^{\infty} \frac{e^{-qx} dx}{\sqrt{1+x}}$	$\ell^q \sqrt{\frac{\pi}{q}}, q > 0$
4	$\int_0^{\infty} \frac{x^2}{\sinh(ax)} dx$	$\frac{4,2}{a^3}$

Таблица 9.3 (продолжение)

№ п/п	Интеграл	Ответ
5	$\int_1^{\infty} \frac{dx}{(a-bx)(x-1)}$	$\frac{\ln \frac{2b-a}{b}}{a-b}$
6	$\int_0^{\infty} x^a \ell^{-ax} dx$	$a^a (a-1)!$
7	$\int_0^{\infty} \frac{x}{\sinh(ax)} dx$	$\frac{\pi^2}{2a^2}$
8	$\int_0^{\infty} x^{2n+1} p \ell^{-px^2} dx$	$\frac{n!}{2p^n}$
9	$\int_0^{\infty} \frac{dx}{1+e^{px}}$	$\frac{\ln 2}{p}$
10	$\int_0^{\infty} \frac{ax dx}{\sinh(ax)}$	$\frac{\pi^2}{2a}$
11	$\int_0^{\infty} e^{-q^2 x^2} dx$	$\frac{\sqrt{\pi}}{2q}$ $q > 0$
12	$\int_0^{\infty} x^n e^{-kx} dx$	$n! k^{-n-1}$ $k > 0$
13	$\int_0^1 \frac{x e^x}{(1+x)^2} dx$	$\frac{e}{2} - 1$
14	$\int_{-\infty}^{+\infty} \frac{e^x}{(\beta + e^x)^2} dx$	$\frac{1}{\beta}$

Таблица 9.3 (окончание)

№ п/п	Интеграл	Ответ
15	$\int_0^{\infty} \frac{x dx}{1-x^3}$	$\frac{2\sqrt{3}}{9}\pi$
16	$\int_0^{\infty} \frac{dx}{(x^2 + a^2)^3}$	$\frac{3\pi}{16a^2}$
17	$\int_{-\infty}^{\infty} \frac{dx}{(ax^2 + 2bx + c)^2}$	$\frac{\pi a}{2(ac - b^2)^{3/2}}$
18	$\int_0^{\infty} \frac{dx}{(x + \sqrt{x^2 + a^2})^2}$	$\frac{2}{3a}$
19	$\int_0^{\infty} \frac{\ell^{-p}}{1+x^3} dx$	$\frac{2\sqrt{3}}{9}\pi \ell^{-3}$
20	$\int_0^{\infty} \frac{x dx}{\sqrt{e^x - 1}}$	$2\pi \ln 2$
21	$\int_0^{\infty} \frac{x e^{-x} dx}{\sqrt{e^x - 1}}$	$\frac{\pi}{2}(2 \ln 2 - 1)$
22	$\int_0^{\infty} \frac{x e^{-2x} dx}{\sqrt{e^x - 1}}$	$\frac{3}{4}\pi \left( \ln 2 - \frac{7}{12} \right)$
23	$\int_0^{\infty} \frac{x^2 e^{-x} dx}{\sqrt{(e^x - 1)^3}}$	$8\pi \ln 2$
24	$\int_0^{\infty} x^{2n+1} e^{-px^2} dx$	$\frac{n!}{2p^{n+1}}$



# Интерполяция — метод моделирования

## 10.1. Этапы компьютерной технологии интерполяции

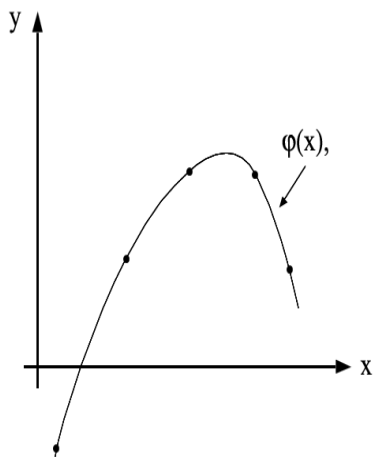
Интерполяцией называется представление функции  $y = f(x)$  функцией  $\varphi(x)$ , идентичной исходной в некоторой области значений аргумента. Задача интерполяции наиболее часто возникает в следующих случаях:

- функция  $y = f(x)$  задана в виде таблицы, полученной в результате эксперимента или обработки статистических данных;
- функция  $y = f(x)$  является сложной аналитической и для упрощения расчетов ее нужно заменить более простой в определенном диапазоне аргументов.

Интерполяция является научной основой моделирования. Она дает возможность разрабатывать математические модели исследуемых объектов или явлений.

Существует два основных вида интерполяции: точная в узлах и приближенная в узлах.

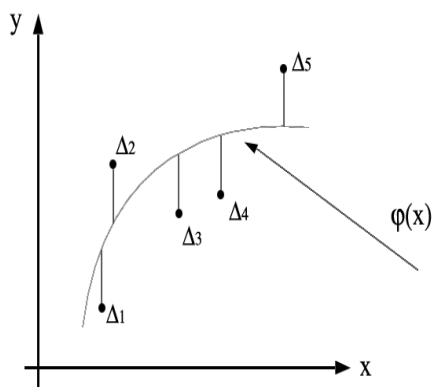
Интерполяцией точной в узлах называется такая интерполяция, результатами которой является функция  $y = \varphi(x)$ . Ее значения совпадают со значениями исходной функции  $y = f(x)$  в узлах интерполяции (рис. 10.1).



**Рис. 10.1.** Интерполяция точная в узлах

На рис. 10.1 точками обозначены табличные значения функции  $y = f(x)$ , а непрерывной кривой — функция интерполяции  $y = \varphi(x)$ . Значения функций в узлах интерполирования совпадают.

Результатами интерполяции приближенной в узлах является функция  $y = \varphi(x)$ , значения которой не совпадают со значениями исходной функции  $y = f(x)$  (рис. 10.2).



**Рис. 10.2.** Интерполяция приближенная в узлах

Такая интерполяция применяется для сглаживания неточностей исходных данных.

Компьютерные технологии интерполяции, независимо от ее вида, состоят из следующих этапов:

- выбор вида функции интерполяции;
- определение коэффициентов функции интерполяции;
- установление адекватности математической модели.

Рассмотрим эти этапы.

### 10.1.1. Выбор вида функции интерполяции

Этот этап является наиболее трудным для учащегося. Существует несколько способов его практической реализации.

#### Способ 1. Представление в виде графика

Функция  $y = f(x)$ , заданная в табличной форме, представляется в виде графика. График сравнивается с графиками типичных функций и на основании сравнения выбирается наиболее подходящая функция. Ниже приводятся часто используемые функции и их графики.

Степенная функция  $y = ax^b$ . График функции показан на рис. 10.3.

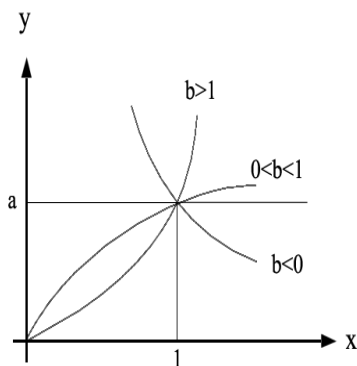
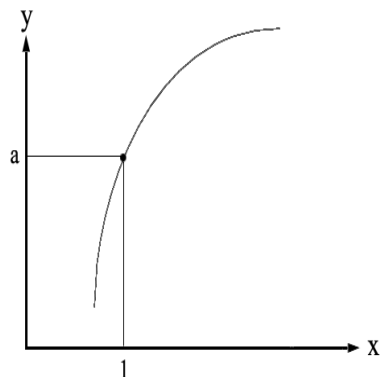


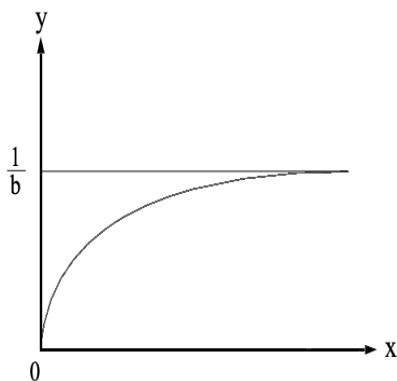
Рис. 10.3. Степенная функция  $y = ax^b$

Логарифмическая функция  $y = a + b \ln(x)$ . График функции показан на рис. 10.4.



**Рис. 10.4.** Логарифмическая функция  $y = a + b \ln(x)$

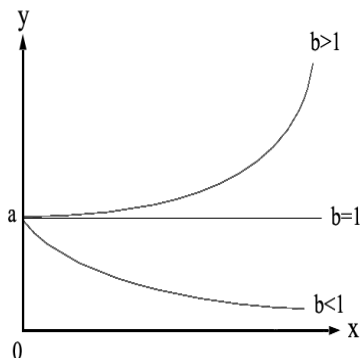
Дробно-линейная функция  $y = \frac{x}{a + bx}$ . График функции показан на рис. 10.5.



**Рис. 10.5.** Дробно-линейная функция  $y = \frac{x}{a + bx}$



*Показательная функция*  $y = ab^x$ . График функции показан на рис. 10.6.



**Рис. 10.6.** Показательная функция  $y = ab^x$

## Способ 2. Линеаризация нелинейных функций

Линеаризация нелинейных функций осуществляется путем представления их в иной системе координат методом замены переменных  $x$  и  $y$ . Покажем способы выравнивания на примерах типичных функций.

*Показательная функция*  $y = ab^x$ .

Выравнивание показательной функции осуществляется путем ее логарифмирования —  $\ln(y) = \ln(a) + x \ln(b)$ . Заменой переменных  $\ln(y) = Y, x = X$  получаем линейную функцию  $Y = \ln(a) + X \ln(b)$ .

*Степенная функция*  $y = ax^b$ .

Представим функцию в виде —  $\ln(y) = \ln(a) + b \ln(x)$ . Тогда введением новых переменных  $\ln(y) = Y, \ln(x) = X$  получим линейную функцию  $Y = \ln(a) + bX$ .

*Логарифмическая функция*  $y = a + b \ln(x)$ .

Выравнивание осуществляется заменой переменных  $y = Y$ ,  $\ln(x) = X$ . После замены переменных получим функцию  $Y = a + bX$ .

*Дробно-линейная функция*  $y = \frac{x}{a + bx}$ .

Представим функцию в виде  $\frac{x}{y} = a + bx$ . Введя новые пере-

менные  $Y = \frac{x}{y}$ ,  $X = x$ , получим линейную функцию  $Y = a + bX$ .

### Способ 3. Анализ табличных разностей

Этот способ применяется в том случае, когда функция интерполяции является многочленом степени  $n$ . Он позволяет выбрать степень многочлена. Если  $n$ -табличные разности одинаковы, то степень многочлена будет не выше  $n$ .

Предположим, что функция интерполяции является многочленом и представлена в виде табл. 10.1.

**Таблица 10.1.** Табличное представление функции интерполяции

$x$	1	2	3	4	5	6	7
$y$	1	8	27	64	125	216	343

В табл. 10.2 приведены значения табличных разностей.

**Таблица 10.2.** Функция и значения ее табличных разностей

$x$	1	2	3	4	5	6	7
$y$	1	8	27	64	125	216	343
$\Delta^{(1)}$	7	19	37	61	91	127	
$\Delta^{(2)}$		12	18	24	30	36	
$\Delta^{(3)}$			6	6	6	6	

Из полиномиальной таблицы следует, что третьи табличные разности постоянны. Тогда интерполяционный полином будет не выше третьей степени, то есть  $y = a_0 + a_1x + a_2x^2 + a_3x^3$ . В данном случае в результате решения задачи полиномиальной интерполяции получим  $y = x^3$ .

#### **Способ 4. Обращение к программам автоматизации интерполяции**

Существуют универсальные программные средства, позволяющие выбрать функцию интерполяции, если исходная функция представлена в табличной форме. Такими программными средствами являются: SIMPLE FORMULA, TableCurve, Curve Expert.

Эти программы выдают более тысячи различных функций с указанием их погрешностей.

#### **Определение коэффициентов функции интерполяции**

Коэффициенты функции интерполяции определяются различными способами, зависящими от ее вида и используемого способа. В системе Derive 5 можно реализовать любой вид интерполяции, путем непосредственных вычислений или используя имеющиеся в системе функции.

## **10.2. Интерполяция точная в узлах**

Интерполяция точная в узлах в системе Derive 5 может быть реализована с помощью следующих трех способов:

- ☐ Решение линейных и нелинейных уравнений (наиболее общий способ интерполяции точной в узлах).
- ☐ Использование функций `POLY_INTERPOLATE` и `LAGRANGE`.
- ☐ Непосредственные вычисления по формулам интерполяции.

Рассмотрим все эти способы.

### 10.2.1. Универсальный метод интерполяции точной в узлах

Решение задачи интерполяции универсальным методом сводится к следующей последовательности действий:

- ☐ выбор вида функции интерполяции;
- ☐ образование системы алгебраических уравнений;
- ☐ решение системы уравнений;
- ☐ проверка адекватности модели.

Технологию решения задач интерполяции методом точным в узлах с помощью системы Derive 5 рассмотрим на примерах.

Предположим, что в результате эксперимента получена функция, представленная в виде табл. 10.3.

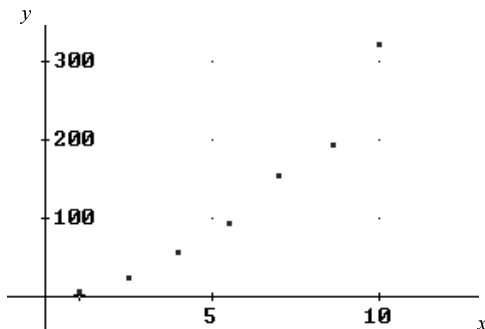
**Таблица 10.3.** Значения исходной функции

$x$	1	2,5	4	5,5	7	8,6	10
$y$	6,5	25	57	93	155	193	321

График функции в виде точек таблицы показан на рис. 10.7.

Из рисунка видно, что функция имеет вид параболы степени не выше второй, то есть функция интерполяции может иметь вид:

$$y = a_0 + a_1x + a_2x^2.$$



**Рис. 10.7.** График исходной функции

Образуем теперь систему уравнений для определения коэффициентов многочлена. Выберем узлы интерполяции, равные 1; 5,5; 10. Тогда система уравнений будет иметь вид:

$$a_0 + a_1 \cdot 1 + a_2 \cdot 1^2 = 6,5;$$

$$a_0 + a_1 \cdot 5,5 + a_2 \cdot 5,5^2 = 93;$$

$$a_0 + a_1 \cdot 10 + a_2 \cdot 10^2 = 321.$$

Технология решения системы уравнений в Derive 5 подробно описана в гл. 7. Решение системы уравнений показано на рис. 10.8.

```
#1:  a0 + a1·x + a2·x2 = s
#2:  a0 + a1·1 + a2·12 = 6.5
#3:  a0 + a1·5.5 + a2·5.52 = 93
#4:  a0 + a1·10 + a2·102 = 321
#5:   $\begin{bmatrix} a0 + a1 \cdot 1 + a2 \cdot 1^2 = 6.5 \\ a0 + a1 \cdot 5.5 + a2 \cdot 5.5^2 = 93 \\ a0 + a1 \cdot 10 + a2 \cdot 10^2 = 321 \end{bmatrix}$ 
#6:  SOLVE  $\left( \begin{bmatrix} a0 + a1 \cdot 1 + a2 \cdot 1^2 = 6.5 \\ a0 + a1 \cdot 5.5 + a2 \cdot 5.5^2 = 93 \\ a0 + a1 \cdot 10 + a2 \cdot 10^2 = 321 \end{bmatrix}, [a0, a1, a2], \text{Real} \right)$ 
#7:  [a0 = 6.493827160 ^ a1 = -3.487654320 ^ a2 = 3.493827160]
```

**Рис. 10.8.** Решение системы уравнений для определения коэффициентов многочлена

В результате решения получен ответ:

$$a_0 = 6,5; a_1 = -3,5; a_2 = 3,5.$$

Тогда функция интерполяции будет иметь вид:

$$\varphi(x) = 6,5 - 3,5x + 3,5x^2.$$

В настоящем примере задача сведена к полиномиальной интерполяции, когда определение коэффициентов функции сводится к решению системы линейных алгебраических уравнений.

Функция интерполяции может быть нелинейной. Тогда решение задачи интерполяции методом точным в узлах сводится к решению системы нелинейных уравнений. Рассмотрим пример такого решения.

Функция  $y = f(x)$  представлена таблицей значений (табл. 10.4).

**Таблица 10.4.** Значения функции

$x$	1	2	3	4	5	6
$y$	0,5	0,57	0,6	0,61	0,62	0,63

Пусть функция интерполяции имеет вид:  $y = \frac{ax}{b + cx}$ .

Неизвестными здесь являются коэффициенты  $a, b, c$ . Их можно определить путем составления и решения следующей системы уравнения:

$$\frac{2a}{b + 2c} = 0,57;$$

$$\frac{4a}{b + 4c} = 0,61;$$

$$\frac{6a}{b + 6c} = 0,63.$$

Решая эту нелинейную систему уравнений методом Ньютона, при начальных значениях:  $a_0 = 1,6$ ;  $b = 1,2$ ;  $c = 2,1$ , получим:  $a = 2$ ;  $b = 1$ ;  $c = 3$ . Тогда функция интерполяции будет иметь вид:

$$y = \frac{2 \cdot x}{1 + 3 \cdot x}.$$

Кроме универсального метода, система Derive 5 имеет две функции интерполяции точной в узлах, которые не требуют от пользователя составления и решения уравнений. Решение задачи осуществляется по данным функции, представленной в виде таблицы.

## 10.2.2. Функция *POLY\_INTERPOLATE*

Эта функция имеет вид:

`POLY_INTERPOLATE(A, x)`,

где  $A$  — матрица функции  $y = f(x)$ , представленной в виде таблицы;  $x$  — аргумент функции  $y = f(x)$ . Сама функция реализует полиномиальную интерполяцию и выдает решение в виде многочлена степени  $(n - 1)$ , где  $n$  — число узлов интерполяции.

Технология интерполяции в среде Derive 5 определяется следующими действиями:

- создание матрицы  $A$  размерности  $m \times n$ , где  $m$  — число строк, равное числу узлов интерполяции;  $n$  — число столбцов (равно в данном случае двум);
- ввод функции `POLY_INTERPOLATE(#к, x)`, где  $\#к$  — номер строки, в которой находится матрица  $A$  (на экране формируется функция с матрицей  $A$ );
- выполнение команды **Simplify**, при этом на экране формируется ответ в виде полинома степени  $(n - 1)$ , в котором числа представлены в форме точных значений, т. е. с числителем и знаменателем (при использовании команды **Approximate** ответом будет многочлен степени  $(n - 1)$ , коэффициентами которого являются числа, представленные в естественной форме).

Рассмотрим еще один пример. Пусть функция  $f(x)$  задана в виде таблицы (табл. 10.5).

**Таблица 10.5.** Таблица значений функции

$x$	1	2	3	4	5
$y$	12	9	7	2	1

Решая задачу интерполяции по описанной выше методике, получим решение, показанное на рис. 10.9.

$$\begin{array}{lcl}
 \#1: & & \begin{vmatrix} 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{vmatrix} \\
 \#2: & POLY\_INTERPOLATE & \left( \begin{bmatrix} 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{bmatrix}, x \right) \\
 \#3: & & \frac{11x^4 - 126x^3 + 493x^2 - 834x + 744}{24} \\
 \#4: & & 0.041666(11x^4 - 126x^3 + 493x^2 - 834x + 744)
 \end{array}$$

**Рис. 10.9.** Уравнения, полученные в процессе решения задачи

### 10.2.3. Функция *LAGRANGE*

Функция *LAGRANGE* имеет вид:

*LAGRANGE* (*A*, *x*),

где *A* — матрица функции  $f(x)$ , представленная в виде таблицы;  
*x* — аргумент функции  $f(x)$ .

Функция осуществляет интерполяцию многочленами степени  $(n - 1)$  по методу Лагранжа.

Технология интерполяции в среде *Derive 5* следующая:

- ❑ вызов утилиты решения задачи интерполяции: **File | Load | Utility | Hermite** (на экране формируется утилита:  
`#1: LOAD (HERMITE.MTH);`
- ❑ ввод матрицы исходных данных (на экране формируется матрица в строке #2);
- ❑ ввод функции *LAGRANGE* (#2, *x*) (на экране формируется команда с матрицей исходных данных, расположенной в строке #2);



- выполнение команды **Simplify** или **Approximate** (на экране формируется ответ, причем в первом случае — в виде точного решения, во втором — с представлением чисел в естественной форме).

Процедуры решения показаны на рис. 10.10.

#1: `LOAD (HERMITE.MTH)`

#2: 
$$\begin{vmatrix} 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{vmatrix}$$

#3: `LAGRANGE` 
$$\left( \begin{vmatrix} 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{vmatrix}, x \right)$$

#4: 
$$\frac{11x^4 - 126x^3 + 493x^2 - 834x + 744}{24}$$

#5: `0.041666666 (11x4-126x3+493x2-834x+744)`

**Рис. 10.10.** Решение задачи интерполяции с использованием функции `LAGRANGE`

В соответствии с технологией полиномиальной интерполяции, реализуемой функцией `POLY_INTERPOLATE`, степень полинома определяется числом узлов, выбираемых пользователем.

## 10.2.4. Сплайн-интерполяция

Сплайн-интерполяция представляет собой интерполяцию функции  $f(x)$  множеством полиномов небольшого порядка на узких участках всего интервала интерполяции. Такие полиномы называют сплайнами. Наиболее часто применяются кубические сплайны. Параметры кубического сплайна выбираются таким образом, чтобы сплайн проходил через три узловые точки. При этом в граничных точках значение сплайна равно табличным значениям функции  $f(x)$  и ее первой и второй производным. Так как график сплайна проходит через узловые точки, то эту интерполяцию можно отнести к методу интерполяции точной в узлах.

Сплайн-интерполяция реализуется в Derive 5 функцией `Spline`. Эта функция имеет вид:

`SPLINE (A) ;`

где  $A$  — матрица функции  $f(x)$ , заданной в виде таблицы.

Технология решения задачи сплайн-интерполяции в среде Derive 5 включает следующие действия:

- ❑ вызов утилиты решения задачи сплайн-интерполяции осуществляется из меню системы: **File | Load | Utility File** с последующим вызовом файла `Spline_s.mth` (на экране формируется строка — `#1: LOAD(SPLINE_S.MTH)`);
- ❑ ввод матрицы  $A$  исходных данных (на экране формируется матрица, например, в строке `#2`);
- ❑ ввод функции `SPLINE(#2)` (на экране формируется команда с матрицей  $A$ );
- ❑ выполнение команды **Simplify** или **Approximate** (на экране формируется ответ, причем в первом случае — в виде точного решения, во втором — в виде приближенного решения с представлением чисел в естественной форме).

## 10.3. Интерполяция приближенная в узлах (аппроксимация)

### 10.3.1. Функция *FIT*

Аппроксимация в Derive 5 реализуется с помощью функции *FIT*, имеющей вид:

*FIT* (A) ;

где A — матрица исходных данных.

Первой строкой матрицы является  $x, \varphi(x)$ ; где  $\varphi(x)$  — функция аппроксимации,  $x$  — аргумент функции  $\varphi(x)$ . Функцией аппроксимации может быть полином, показательная, степенная, логарифмическая и другие функции (по выбору пользователя).

Вслед за строкой  $x, \varphi(x)$  следуют табличные данные исходной функции  $y = f(x)$ . Пусть, например, функция  $y = f(x)$  задана в виде таблицы (табл. 10.6).

**Таблица 10.6.** Исходная функция интерполяции

$x$	1	3	7	12	18
$y$	3,2	7	9,65	11,3	16,8

И пусть функцией интерполяции является полином третьей степени  $y = a_0 + a_1x + a_2x^2 + a_3x^3$ . Тогда матрица *A* будет иметь вид, показанный в форме табл. 10.7.

**Таблица 10.7.** Матрица функции *FIT*

$x$	$y = a_0 + a_1x + a_2x^2 + a_3x^3$
1	3,2
3	7
7	9,65
12	11,3
18	16,8

Аппроксимация с помощью функции `FIT` осуществляется по методу наименьших квадратов.

Технология аппроксимации в среде `Derive 5` включает следующие действия:

- ❑ ввод матрицы  $A$  (на экране, в строке #1, формируется матрица);
- ❑ ввод команды — `FIT #1` (на экране формируется команда `FIT` с матрицей данных);
- ❑ выполнение команды **Approx** (на экране формируется ответ в виде формулы).

Рассмотрим пример. Пусть функция  $y = f(x)$  задана в виде табл. 10.8.

**Таблица 10.8.** Табличное представление исходной функции интерполяции

$x$	1	2	3	4
$y$	2	4	8	12

Необходимо определить функцию интерполяции, если интерполяционный полином-многочлен второй степени  $y = a + bx + cx^2$ .

Тогда после ввода исходных данных на экране будет матрица  $A$  следующего вида:

$$\#1: \begin{vmatrix} x & a + bx + cx^2 \\ 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{vmatrix}$$

Набор в строке пользователя функции  $y = \text{FIT}(\#1)$ . После нажатия клавиши <Enter> на экране появится функция в следующем виде:

$$\#2: \quad y = \text{FIT} \left| \begin{array}{cc} x & a + bx + cx^2 \\ 1 & 2 \\ 2 & 4 \\ 3 & 8 \\ 4 & 12 \end{array} \right|$$

Команда **Approximate** — на экране ответ  $y = 0,5x^2 + 0,9x + 0,5$ .

## Замечания о команде *FIT*

- Если число неизвестных функции интерполяции равно числу строк исходной таблицы данных, то с помощью команды **FIT** решается задача интерполяции точной в узлах. Если число неизвестных меньше числа строк исходной таблицы, то решается задача интерполяции приближенной в узлах (аппроксимации). Если же число неизвестных больше числа строк таблицы, то задача не решается.
- Функция аппроксимации практически может быть любого вида, при этом в аргументе функции не должно быть искомым переменных. Например, функция  $y = a + b \ln(x)$  вполне может быть аппроксимирующей, а функция  $y = a + b \ln(cx)$  нет, так как произведение  $cx$  программой воспринимается как аргумент.
- Команда **FIT** позволяет решать задачи аппроксимации функций многих переменных.
- Например, пусть функция задана в виде табл. 10.9.

**Таблица 10.9.** Задание функции в табличном виде

$x$	1	2	4	8	14	20
$y$	2,3	6,1	7,4	9	12	16,5
$z$	12,4	8	7,2	5,4	3,1	1,9
$w$	2,5	4,6	7,4	10,2	13,8	15,6

И пусть функция  $w(x, y, z)$  является линейной, тогда команда FIT будет иметь вид:

	x	y	z	$a + bx + cy + dz$
	1	2, 3	12, 4	2, 5
	2	6, 1	8	4, 6
FIT	4	7, 4	7, 2	7, 4
	8	9	5, 4	10, 2
	14	12	3, 1	13, 8
	20	16, 5	1, 9	15, 6

Если теперь функцию  $w(x, y, z)$  принять за квадратичную, то следует изменить лишь первую строку таблицы, т. е. вместо линейной функции записать следующую квадратичную:

$$a + b_1x + b_2x^2 + c_1y + c_2y^2 + d_1z + d_2z^2 + e_1xy + e_2xz + e_3yz.$$

Правда, в этом случае, наша задача не будет решена, так как число неизвестных функции аппроксимации больше числа узлов интерполяции (числа строк таблицы). Для решения необходимо иметь не менее 10 строк исходной таблицы (по числу коэффициентов квадратичной функции).

### 10.3.2. Функция ALL\_SEVEN

Функция ALL\_SEVEN имеет вид:

ALL\_SEVEN(A),

где A — матрица исходных данных. Она позволяет интерполировать функцию, заданную в виде таблицы, пятью функциями вида:

□ линейная —  $y = a + bx$ ;

□ показательная —  $y = ae^{bx}$ ;

□ степенная —  $y = ax^b$ ;

□ логарифмическая —  $y = a \ln(x) + b$ ;

□ дробно-линейная —  $y = a + b/x$ ,  $y = a/(b + cx)$ ,  $y = ax/(b + cx)$ .

Технология решения задачи интерполяции в среде Derive 5 определяется следующими действиями:

обращение к утилите решения задачи интерполяции — **File | Load | Utility File | Curv\_fit** (после нажатия кнопки **Открыть** на экране формируется выражение — `#1:LOAD(Curv_fit.mth);`

□ ввод матрицы исходных данных (на экране, в строке #2, формируется матрица);

□ ввод команды **ALL\_SEVEN** (#2) (на экране формируется команда с матрицей исходных данных);

□ выполнение команды **Approximate** (на экране формируется ответ в виде семи формул).

Рассмотрим пример. Пусть функция задана в виде табл. 10.10.

**Таблица 10.10.** Задание функции в табличном виде

$x$	1	2	3	4	5	6
$y$	6,5	20	53,5	167	473	1470

После исполнения всех перечисленных выше команд на экране будет сформирован ответ в следующем виде:

$$251.142t - 514$$

$$2.21689e^{1.07812t}$$

$$3.83192t^{2.94214}$$

$$591.156 \ln t - 283.227$$

$$763.015 - \frac{974.732}{t}$$

$$4.07641 \cdot 10^{-7}$$

$$\frac{1.18175 \cdot 10^6 - 1.94317 \cdot 10^5 t}{}$$

$$\frac{1.28619 \cdot 10^7 t}{1.27441 \cdot 10^6 - 2.06659 \cdot 10^5 t}$$

В полученном ответе аргумент функции обозначен символом  $t$ , вместо привычного  $x$ .

Следует иметь в виду, что некоторые из полученных формул могут иметь большие погрешности, хотя все они получены на основании одного и того же метода — метода наименьших квадратов.

### Примечание

Окончательный выбор формулы аппроксимации остается за пользователем.

Функция `ALL_SEVEN` полезна в тех случаях, когда пользователь не знает вида функции, а поэтому не может воспользоваться командой `FIT`.

## 10.3.3. Паде-аппроксимация

Паде-аппроксимация предназначена для аппроксимации нелинейных функций дробно-рациональными функциями вида:

$$y(x) = \frac{b_0 + b_1 x + b_2 x^2 + \dots + b_m x^m}{a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n}.$$

Метод является одним из наиболее точных. Его реализация в `Derive 5` возможна лишь в том случае, если нелинейная функция задана в виде аналитического выражения.

Функция Паде имеет вид:

`PADE(y, x, x0, m, 10n)`,

где  $y, x$  — функция и ее аргумент;  $x_0$  — значение аргумента, вблизи которого обеспечивается высокая точность аппроксимации;  $m$  — степень многочлена числителя;  $n$  — степень многочлена знаменателя.

Рассмотрим пример использования данной функции.



Пусть нелинейная функция имеет вид:  $y(x) = 2e^{-1.2x} \cos(x)$ . Необходимо функцию аппроксимации  $\phi(x)$  представить в виде дробно-рациональной функции вблизи значения аргумента  $x = 0$  при  $m = 2, n = 3$ .

Последовательность решения задачи аппроксимации, с использованием функции Паде, в среде Derive 5 имеет следующий вид:

□ загрузка файла утилиты решения задачи аппроксимации Паде — **File | Load | Utility | Approx.mth** (на экране формируется — `Load(Approx.mth)`);

□ ввод функции — `PADE(2*EXP(-1.2*x)*cos(x), x, 0, 2, 3)`;

□ выполнение команды **Simplify** или **Approximate**.

Откликом будет следующая дробно-рациональная функция:

$$-\frac{10(679...x^2 - 848...x - 115)}{161...x^3 + 239...x^2 + 493...x + 576...}$$

## 10.4. Решение задач интерполяции путем прямых вычислений

Существует большое число методов интерполяции, которые не реализованы в Derive 5 с помощью функций и команд. К ним относятся: метод Ньютона, Гаусса, Стирлинга, Бесселя и др. Все эти и другие методы могут быть реализованы весьма эффективно с помощью функций и команд системы Derive 5.

Технологию прямых вычислений покажем на примерах.

Интерполяционная формула Ньютона имеет вид:

$$\begin{aligned} y_n(x) = & y_0 + \frac{\Delta y_0}{1!h}(x - x_0) + \frac{\Delta^2 y_0}{2!h^2}(x - x_0)(x - x_r) + \\ & + \frac{\Delta^3 y_0}{3!h^3}(x - x_0)(x - x_1)(x - x_2) + \dots \\ & \dots + \frac{\Delta^n y_0}{n!h^n}(x - x_0)(x - x_1) \dots (x - x_{n-1}). \end{aligned}$$

В ней приняты следующие обозначения:

- $x_i, i = 0, 1, \dots, (n-1)$  — узлы интерполяции с постоянным шагом  $h$ , т. е.  $x_1 = x_0 + h, x_2 = x_1 + h, \dots, x_{n-1} = x_{n-2} + h$ ;
- $y_0$  — значение функции при  $x = x_0$ ;
- $\Delta^k y_0, k = 1, 2, \dots, n$  — табличные разности (первая, вторая, ...,  $n$ -я);
- $n$  — степень интерполяционного полинома.

Необходимо найти коэффициенты интерполяционного полинома, если функция  $y = f(x)$  задана в виде табл. 10.5.

Так как число узлов равно пяти, то интерполяционный полином будет не выше четвертой степени. Пусть  $y(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$ , тогда вычисления по формуле Ньютона можно легко реализовать в системе Derive 5, если известны табличные разности. Получим их путем очевидных вычислений, которые в данном примере не требуют компьютерных технологий. Результаты вычислений приведены в табл. 10.11.

**Таблица 10.11.** Табличные разности функции интерполяции

$x$	$y$	$\Delta y$	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$
1	12				
2	9	-3			
3	7	-2	1		
4	2	-5	-3	-4	
5	1	-1	4	7	11

Технология решения задачи в системе Derive 5 определяется следующей последовательностью действий:

- набор и ввод формулы Ньютона (на экране формируется формула в строке #1);
- подстановка в формулу, с помощью команды **Sub**, значений  $x_i, y_0, \Delta y^i$ , взятых из таблицы 10.10 (на экране формируется формула с численными значениями переменных);

- упрощение формулы с помощью пункта главного меню **Simplify** | **Factor**.

В результате проделанных действий, на экране будет сформирован ответ в виде следующего полинома:

$y = 31 - 34,75x + 20,514x^2 - 5,25x^3 + 0,458333x^4$ , что совпадает с ответом, полученным ранее при решении задачи с использованием функции `POLY_INTERPOLATE`.

При подстановке в формулу Ньютона переменных, табличные разности берутся из первой косой строки табл. 10.10, то есть  $\Delta y_0 = -3$ ,  $\Delta^2 y_0 = 1$ ,  $\Delta^3 y_0 = -4$ ,  $\Delta^4 y_0 = 11$ .

Решить задачу полиномиальной интерполяции по методу Ньютона можно проще, не вычисляя табличных разностей.

Исходная формула Ньютона для нашего примера имеет вид:

$$y(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + c_3(x - x_0)(x - x_1)(x - x_2) + c_4(x - x_0)(x - x_1)(x - x_2)(x - x_3).$$

Тогда интерполяционный полином можно получить с помощью функции `FIT`.

Технология решения задачи с помощью `Derive 5`:

- ввод исходной формулы Ньютона (на экране, в строке #1, формируется формула);
- подстановка в формулу, с помощью команды **Sub** панели инструментов, значений  $x_0 = 1, x_1 = 2, x_2 = 3, x_3 = 4$  (на экране, в строке #2, формируется формула с численными значениями узлов интерполяции);
- набор и ввод следующей матрицы:

$$\#3: \begin{vmatrix} x & y \\ 1 & 12 \\ 2 & 9 \\ 3 & 7 \\ 4 & 2 \\ 5 & 1 \end{vmatrix}$$

- ❑ набор и ввод команды `FIT (#3)` (на экране формируется матрица с функцией `FIT`);
- ❑ выполнение команды **Simplify | Factor** (на экране формируется ответ, совпадающий с прежним).

Аналогично приведенному примеру решаются задачи интерполяции по методу Ньютона в случае переменного шага, а также методами Гаусса, Стирлинга, Бесселя и др.

## 10.5. Проверка адекватности модели

Качественно проверку адекватности модели можно установить путем сравнения графиков исходной функции  $f(x)$ , представленной в табличной форме, и функции интерполяции  $\varphi(x)$ . Если точки дискретной функции  $f(x)$  находятся на кривой функции  $\varphi(x)$ , то можно утверждать, что адекватность модели доказана.

Рассмотрим этот способ на примерах.

В примере, использующим функцию `POLY_INTERPOLATE` (см. разд. 10.2.2), получена следующая функция интерполяции:

$$\varphi(x) = 0,458333x^4 - 5,25x^3 + 20,5146x^2 - 34,75x + 31.$$

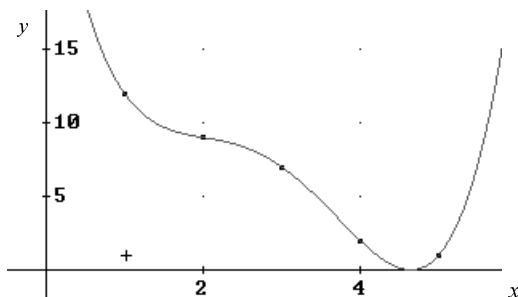
Эта функция является математической моделью объекта, опытные данные о котором представлены в табл. 10.10.

На рис. 10.11 показаны функции  $f(x)$  и  $\varphi(x)$ , полученные с помощью команды **2D-plot**.

Из рисунка видно, что точки функции  $f(x)$  лежат на кривой  $\varphi(x)$ . Адекватность модели доказана.

### Примечание

Этого следовало ожидать, так как для решения задачи методом Ньютона использован полином четвертой степени и интерполяция точная в узлах.



**Рис. 10.11.** Графики функций  $f(x)$  и  $\varphi(x)$

Выполним аппроксимацию Паде функции  $y(x) = e^{-2x} \cos x + 1$

дробно-рациональной функцией  $y(x) = \frac{b_0 + b_1 + b_2 x^2}{a_0 + a_1 x + a_2 x^2 + a_3 x^3}$

в диапазоне изменения аргумента  $x$  от  $-2$  до  $2$ . Пусть  $x_0 = 0$ . Тогда функция Паде будет иметь вид – PADE  $(e^{-2x} \cos x + 1, x, 0, 2, 3)$ , а ее откликом будет искомая функция  $\varphi(x)$ . Процедуры решения задачи на экране имеют следующий вид:

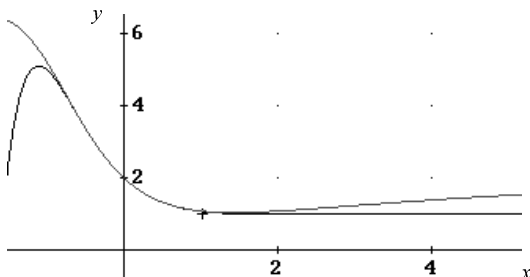
#1: LOAD (APPROX. MTH)

#2:  $e^{-2x} \cos(x) + 1$

#3: PADE  $(e^{-2x} \cos(x) + 1, x, 0, 2, 3)$

#4: 
$$-\frac{24(193x^2 - 149x + 470)}{11x^3 - 1938x^2 - 3852x - 5640}$$

Функции  $f(x)$  и  $\varphi(x)$  представлены на рис. 10.12.



**Рис. 10.12.** Функция интерполяции Паде

Из рисунка видно, что в диапазоне изменения аргумента от  $-1$  до  $2$  функции практически совпадают.

Пусть функция  $y = f(x)$  представлена в виде табл. 10.12.

**Таблица 10.12.** Табличное представление функции для случая полинома второй степени

$x$	1	2	3	4	5	6	7
$y$	1,1	3,7	9,5	15	28	30	50

Необходимо найти функцию интерполяции, если  $\varphi(x) = a_0 + a_1x + a_2x^2$ .

Воспользуемся функцией `FIT(A)`. В результате ее реализации программа `Derive 5` выдала следующее решение:  $\varphi(x) = 0,35714 - 0,12619x + 0,9881x^2$ . Процедуры вычислений на экране показаны на рис. 10.13 и 10.14.

Из рисунка видно, что точки функции  $f(x)$  не лежат на кривой  $\varphi(x)$ , хотя и расположены достаточно близко к ней. В этом случае вряд ли можно на основании графика сделать вывод об адекватности модели. Очевидно, что для доказательства адекватности модели необходимо вычислить погрешность интерполяции.

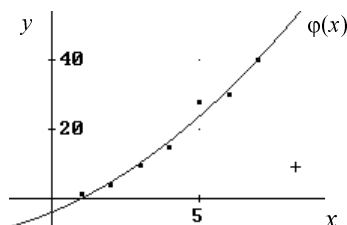
$$\#1: \begin{bmatrix} x & a1 + a2 \cdot x + a3 \cdot x^2 \\ 1 & 1.1 \\ 2 & 3.7 \\ 3 & 9.5 \\ 4 & 15 \\ 5 & 28 \\ 6 & 30 \\ 7 & 40 \end{bmatrix}$$

**Рис. 10.13.** Полиномиальная интерполяция функции, заданной табл. 10.11

$$\#2: \quad \text{FIT} \quad \begin{bmatrix} x & a1 + a2 \cdot x + a3 \cdot x^2 \\ 1 & 1.1 \\ 2 & 3.7 \\ 3 & 9.5 \\ 4 & 15 \\ 5 & 28 \\ 6 & 30 \\ 7 & 40 \end{bmatrix}$$

$$\#3: \quad 0.3928571428 \cdot x^2 + 3.564285714 \cdot x - 3.928571428$$

**Рис. 10.14.** Решение полиномиальной интерполяции функции, заданной табл. 10.11



**Рис. 10.15.** График функции интерполяции значений из табл. 10.11

Графики функций  $f(x)$  и  $\varphi(x)$  показаны на рис. 10.15.

Наиболее часто за критерий точности интерполяции принимают величину абсолютной  $\varepsilon$  и максимальной относительной  $\delta$  погрешностей. Абсолютная среднеквадратическая погрешность вычисляется по формуле:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^n \Delta_i^2}{n}}, \quad (10.1)$$

где  $n$  — число узлов функции  $f(x)$ ;  $\Delta_i = |f(x_i) - \varphi(x_i)|$  — абсолютное значение разности исходной функции и функции интерполяции в узле  $i$ ,  $i = 1, 2, \dots, n$ .

Максимальная относительная погрешность вычисляется по формуле:

$$\delta_{\max} = \frac{\varepsilon}{|f_{\max}(x)|} \cdot 100\%, \quad (10.2)$$

где  $f_{\max}(x)$  — максимальное значение функции  $f(x)$ .

Покажем методику вычисления погрешностей на примере.

Компьютерная технология определения погрешностей  $\varepsilon$  и  $\delta_{\max}$  в системе Derive 5 состоит в выполнении следующих действий.

□ Образование векторов функций  $f(x)$  и  $\varphi(x)$ .

Так как функция  $f(x)$  задана в виде таблицы, то ее представление в виде вектора  $y(x)$  осуществляется по данным таблицы с помощью команды **Author Vector**. Вектор функции  $\varphi(x)$  образуется путем ее табулирования. Для нашего примера функция табуляции имеет вид:

`VECTOR (#1, x, 1, 7);`

где #1 — номер экранной строки, в которой находится функция аппроксимации  $\varphi(x)$ . В результате реализации этой функции получим вектор, расположенный в строке #3.

□ Образование суммы квадратов разностей векторов  $f(x)$  и  $\varphi(x)$ .

Предположим, что вектор функции  $f(x)$  находится на экране в строке #4. Тогда квадрат вектора разностей образуется вводом выражения —  $(\#3 - \#4)^2$  и последующим исполнением команды

**Approximate**. В результате получим значение  $\sum_{i=1}^n \Delta_i^2$  в виде числа.

В нашем случае это будет:

$$\sum_{i=1}^7 \Delta_i^2 = 44,9238.$$

Предположим, что данное число находится в экранной строке — #6.



# □ Вычисление погрешностей.

В соответствии с формулой (10.1) необходимо набрать и ввести выражение — `sqrt(#6/7)`. После щелчка по кнопке **Approximate**, расположенной на панели инструментов, получим значение абсолютной среднеквадратической погрешности  $\varepsilon$ . Тогда максимальная и минимальная относительные погрешности будут иметь значения:

$$\delta_{\max} = \frac{\varepsilon}{y_{\max}(x)} \cdot 100\%, \quad \delta_{\min} = \frac{\varepsilon}{y_{\min}(x)} \cdot 100\%.$$

Вычисление погрешностей можно получить непосредственно по формуле (10.1) без вычисления значения суммы квадратов разностей.

На рис. 10.16 приведены процедуры вычисления погрешностей, которые отображаются на экране.

```

#1:      0.3928571428·x2 + 3.564285714·x - 3.928571428
#2:  [1.1, 3.7, 9.5, 15, 28, 30, 40]
#3:  VECTOR(0.3928571428·x2 + 3.564285714·x - 3.928571428, x, 1, 7)
#4:  [0.02857142857, 4.771428571, 10.3, 16.61428571, 23.71428571, 31.6, 40.27142857]
#5:  ([1.1, 3.7, 9.5, 15, 28, 30, 40] - [0.02857142857, 4.771428571, 10.3, 16.61428571,
      23.71428571, 31.6, 40.27142857])2
#6:      26.54285714
#7:  √(  $\frac{26.54285714}{7}$  )
#8:      1.947263909
#9:   $\frac{1.947263909}{1.1} \cdot 100$ 
#10:      177.0239917
#11:   $\frac{1.947263909}{40} \cdot 100$ 
#12:      4.868159772

```

**Рис. 10.16.** Вычисление погрешностей интерполяции

Из выполненных расчетов можно сделать вывод о том, что полученная математическая модель не является адекватной исследуемому объекту, так как ее максимальная относительная погрешность слишком велика (177%). Причинами отрицательного результата табулирования могут быть:

- неудачно выбрана функция интерполяции;
- низкая точность исходных данных.

# Глава 11



## Задачи повышенной сложности

Задачами повышенной сложности будем называть такие, которые для своего решения требуют выполнения совокупности математических операций. Такие задачи позволяют более глубоко и всесторонне изучить математическую систему, в данном случае Derive 5. Трудность решения этих задач состоит в том, что учащийся должен самостоятельно разработать последовательность выполнения математических операций, а для этого он должен хорошо усвоить математические функции системы Derive 5 и технологии их реализации.

Задачи повышенной сложности полезно решать всем, кто самостоятельно изучает математическую систему. Они могут пригодиться при приеме зачетов, выполнении домашних заданий и лабораторных работ. Их можно рекомендовать и студентам заочной формы обучения.

### Задача 11.1

Функция  $y = f(x)$  задана в виде табл. 11.1.

**Таблица 11.1.** Табличное задание функции

$x$	0,2	0,5	1	1,3	1,7	2	2,2	2,5
$y$	4,564	7,225	11,7	13,88	15,53	15,5	14,484	11,625

Найти:

□ координаты максимума функции  $y = f(x)$ ;

□ значение интеграла  $\int_1^2 f(x)dx$ ;

□ функцию  $y = \int f(x)dx$  в виде таблицы для значений  $x$ , приведенных в табл. 11.1.

Ответы:  $x_{\max} = 1,8244$ ;  $y_{\max} = 15,64$ ;  $\int_1^2 f(x)dx = 14,47$ .

### Примечание

Ответы могут незначительно отличаться от полученных решений, так как они зависят от вида функции интерполяции.

## Задача 11.2

Функция  $y = f''(x)$  задана в виде табл. 11.2.

**Таблица 11.2.** Табличное представление второй производной функции

$x$	0,5	1	1,5	2	2,5	3	3,5	4	4,5	5
$y$	2,3	11,35	28,85	55,9	93,4	142	202,5	275,3	361	460

Найти:

- функцию  $y = f'(x)$  и представить ее в виде формулы, таблицы и графика;
- найти корни уравнения  $f(x) - 1 = 0$ ;
- найти  $\lim_{x \rightarrow 1} f(x)$  при  $x \rightarrow 1$ .

Ответы:

- корни уравнения  
 $x_{1,2} = 1,05$ ;  $x_{3,4} = 0,115 \pm 1,047j$ ;  $x_{5,6} = -1,028 \pm 0,229j$ ;
- $\lim_{x \rightarrow 1} f(x) = 0,8$ ;

- функцию  $y = f(x)$  выбрать лучшую из следующих:  $y = a \ln x + b$ ,  $y = ax^b$ ,  $y = \frac{a}{b + cx}$ .

## Задача 11.3

Функция  $y = f(x)$  задана в виде табл. 11.3.

**Таблица 11.3.** Табличное представление функции  $y = f(x)$

$x$	1	2	3	4	5	6	7	8	9	10
$y$	1,4	3,13	4,15	4,87	5,42	5,88	6,26	6,6	6,89	7,156

Необходимо:

- представить в виде формулы, таблицы и графика вторую производную функции  $f(x)$ ;
- найти:
- $\lim_{x \rightarrow 1} f'(x)$ ;
  - $\int f(x) dx, \int_1^5 f(x) dx$ .

Ответы:

- $\lim_{x \rightarrow 1} f'(x) = 2,5$ ;
- $\int f(x) dx = 2,5x \ln x - 1,1x$ ;  $\int_1^5 f(x) dx = 15,72$ .
- Функция  $f(x)$  может принадлежать к одной из следующих:  $y = \frac{a}{b + cx}$ ,  $y = ae^{bx}$ ,  $y = a \ln x + b$ .

## Задача 11.4

Функция  $y = f(x)$  задана в виде табл. 11.4.

**Таблица 11.4.** Табличное задание функции для решения задачи интерполяции

$x$	1	2	3	4	5	6	7	8	9	10
$y$	-5	12,5	2,78	1,56	1,09	0,83	0,68	0,57	0,49	0,43

Необходимо:

- ☐ найти оптимальную функцию интерполяции;
- ☐ вычислить погрешность функции интерполяции;
- ☐ найти корни уравнения  $f(x) + x - 1 = 0$ .

Функция интерполяции может иметь вид одной из следующих функций:

- ☐  $y = a + bx$ ;
- ☐  $y = ae^{bx}$ ;
- ☐  $y = ax^b$ ;
- ☐  $y = a \ln x + b$ ;
- ☐  $y = a + \frac{b}{x}$ ;
- ☐  $y = \frac{a}{b + cx}$ ;
- ☐  $y = \frac{ax}{b + cx}$ .

Ответ:

- ☐ корни уравнения  $x_{1,2} = 1,357 \pm 1,856j$ .

## Задача 11.5

Корнями уравнения  $ax^3 + bx^2 + cx + d = 0$  являются  $x_1 = 1,5$ ;  $x_2 = -2 + 3j$ .

Необходимо:

- ☐ найти функцию  $f(x)$ , определив значения коэффициентов  $a, b, c, d$ ;
- ☐ представить функцию в виде таблицы и графика;
- ☐ найти корни уравнения  $\frac{f(x)}{x} - \ln 2x - 3 = 0$ .

Ответы:

- ☐  $a = 1, b = 2,5, c = 7, d = 19,5$ ;
- ☐ корнем уравнения является  $x_1 = 4,3$ .

## Задача 11.6

Дана функция:  $f(x) = ax^3 + bx^2 + cx + 3,5$ .

Известно, что корнями уравнений  $f(x) = 0$ ,  $f'(x) = 0$  и  $f''(x) = 0$  являются соответственно  $x_1 = 0,774$ ;  $x_2 = 0,515$ ;  $x_3 = 0,58$ .

Найти:

- ☐ функцию  $f(x)$ , вычислив коэффициенты  $a, b, c$ ;
- ☐ представить функцию  $f(x)$  в табличном и графическом виде;
- ☐ определить корни уравнения  $f(x) = 0$ .

Ответы:

- ☐ значения коэффициентов:  $a = 7,04$ ;  $b = 12,26$ ;  $c = -18,23$ ;
- ☐ корни уравнения:  $x_1 = 0,234$ ;  $x_2 = 0,774$ ;  $x_3 = 2,75$ .

## Задача 11.7

Корнями уравнения  $f(x) = 0$  являются:  $x_1 = 1,5; x_{2,3} = 3,1 \pm 1,6j$ .

Определить:

☐ корни уравнения  $f(x) + x - 1 = 0$ ;

☐ производную функции  $f(x)$ ;

☐ значение интеграла  $\int_1^{10} f(x) dx$ ;

☐ представить функцию в виде таблицы и графика.

Предполагается, что функция  $f(x)$  является полиномом третьей степени.

Ответы:

☐ корнями уравнения являются:  $x_1 = 1,42; x_{2,3} = 3,14 \pm 1,92j$ ;

☐ значение интеграла  $\int_0^{10} f(x) dx = 834,12$ .

## Задача 11.8

Функция  $y = f(x)$  представлена в виде табл. 11.5.

**Таблица 11.5.** Табличное представление функции для решения задачи интерполяции

$x$	3	4	5	6	7	8	9	10
$y$	0,336	1,05	1,9	2,9	4,09	5,5	7,17	9,158

Необходимо:

☐ решить задачу интерполяции, если известно, что функция интерполяции имеет вид:  $y = ab^{cx} - 3,2$ ;



- ☐ вычислить абсолютную и максимальную относительную среднеквадратические погрешности модели;
- ☐ найти значение интеграла  $\int_1^8 f(x)dx$ ;
- ☐ найти корни уравнения  $f(x) = 0$ ;
- ☐ представить производную  $\frac{dy(x)}{dx}$  в аналитической, табличной и графической формах.

Ответы:

- ☐  $\int_1^8 f(x)dx = 12,28$ ;
- ☐ корни уравнения:  $x_1 = 2,46$ ;  $x_{2,3} = 2,46 \pm 39,84j$ .

## Задача 11.9

Опытные данные эксперимента приведены в табл. 11.6.

**Таблица 11.6.** Табличные данные эксперимента

$x_1$	1	2,3	3,4	5	6,3	7,9	11,7	14,5	17
$x_2$	0,5	0,95	1,75	3,06	4,27	3,4	1,95	1	0,35
$y$	5,375	22,25	48	106,5	174,27	244,95	515,85	798,75	1106,96

Решить задачу многопараметрической интерполяции, предполагая, что функция  $y = f(x_1, x_2)$  является квадратической. Определить погрешности математической модели.

Представить функцию  $y = f(x_1, x_2)$  в виде таблицы для значений  $x_1 = 1, 2, \dots, 10$ ;  $x_2 = 1, 2, \dots, 10$ ; выполнив табулирование функции по двум параметрам.

## Задача 11.10

Решить следующую систему дифференциальных уравнений методом Рунге-Кутты:

$$\frac{dy_0(t)}{dt} = -ay_0(t) + my_1(t);$$

$$\frac{dy_1(t)}{dt} = ay_0(t) - (a+m)y_1(t) + 2my_2(t);$$

$$\frac{dy_2(t)}{dt} = ay_1(t) - 2my_2(t).$$

Начальные условия:  $y_0(0) = 1$ ,  $y_1(0) = y_2(0) = 0$ .

По данным решения системы дифференциальных уравнений получить функцию  $y_0(t)$  в аналитическом виде, решая задачу интерполяции. Использовать для получения функции  $y_0(t)$  два метода интерполяции: точный в узлах и приближенный в узлах с помощью функции `FIT` системы Derive 5.

Сравнить результаты, вычислив погрешности моделей.

Коэффициентами  $a$  и  $m$  системы дифференциальных уравнений являются числа:  $a = 0,5 \cdot 10^2$ ,  $m = 1$ .

Рекомендации по решению задачи:

- ☐ функцией интерполяции является полином третьей степени  $y_0(t) = a + bt + ct^2 + dt^3$ ;
- ☐ диапазон изменения  $t$  взять по своему желанию.

## Задача 11.11

Создать собственную пользовательскую функцию вида:

$$y(x) = \frac{e^{ax}}{x} + bx^2 - cx + d.$$

Обращаясь к ней многократно, определить:

- ☐ корни уравнения  $y(x) = 0$  при  $a = 1,2$ ;  $b = 2,4$ ;  $c = 0,5$ ;  $d = 3,5$ ;
- ☐ первую и вторую производные в аналитическом виде;
- ☐ значение интеграла  $\int_1^{18} y(x) dx$ ;
- ☐ выполнить табулирование функции  $y(x)$  при двух диапазонах изменения аргумента:  $x = 1, 2, 3, \dots, 10$  и  $x = 0,5; -1; -1,5; \dots; -7$ .

## Задача 11.12

Даны следующие три функции:  $y_1(x) = e^{ax} \cos x$ ,  
 $y_2(x) = a + b \ln cx$ ,  $y_3 = \arctg 0,5x + x + 1$ .

Необходимо:

- ☐ табулировать функции в диапазоне  $x = 1; 1,5; 2; \dots; 5,5$  с помощью одной функции табуляции;
- ☐ по данным таблицы найти в табличном виде сумму  $y_1(x) + y_2(x) + y_3(x)$ ;
- ☐ решить задачу интерполяции, предполагая, что функция интерполяции является многочленом степени  $n$ .

Степень многочлена выбрать, воспользовавшись графическим способом или с помощью табличных разностей.

## Задача 11.13

Разложить функцию  $e^x$  в степенной ряд в диапазоне  $x$  от 0 до 1, ограничившись тремя, пятью и десятью членами.

Вычислить погрешность полученных функций путем их табулирования и вычисления среднеквадратических погрешностей.

## Задача 11.14

Даны две следующие функции:

$$y_1(x) = e^{-2x} - 0,5, \quad y_2(x) = 2^x - 3x + 1.$$

Найти:

- ☐ координаты точек пересечения, которые являются корнями  $x_1, x_2, x_3$  уравнения  $y_1(x) - y_2(x) = 0$ ;
- ☐ найти корни и образовать полином третьей степени, корни которого будут равны  $x_1, x_2, x_3$ ;
- ☐ представить функцию (полином третьей степени) в табличном и графическом виде и определить значение интеграла

$$\int_0^{10} f(x) dx.$$

Ответы:

- ☐ полином третьей степени  $f(x) = x^3 - 3,33x^2 + 0,61x + 2,61$ ;
- ☐ значение интеграла  $\int_0^{10} f(x) dx = 36,1$ .

## Задача 11.15

Корнями уравнения  $f(x) = 0$  являются:  $x_1 = -2,5$ ;  $x_2 = -1,3$ ;  $x_3 = 7$ .

Необходимо:

- ☐ найти функцию  $f(x)$ , предполагая, что  $f(x)$  является многочленом третьей степени;
- ☐ вычислить значение производной в точках  $x_1, x_2, x_3$ ;
- ☐ определить значение интеграла  $\int_0^{20} f'(x) dx$ ;
- ☐ найти координаты особых точек (max, min, точек перегиба).

Ответы:

$$f'(-2,5) = 11,4; f'(-1,3) = -9,96; f'(7) = 78,85;$$

$$\int_0^{20} f'(x) dx = 6253.$$

## Задача 11.16

Корнями уравнения  $xe^{1,2x} - e^{1,2x} + xe^{-2,2x} - e^{-2,2x} = 0$  являются:

$$x_1 = 1, \quad x_{2,3} = \pm \frac{15\pi}{17}j, \quad x_{4,5} = \pm \frac{5\pi}{17}j.$$

Необходимо найти многочлен пятой степени, предполагая, что найденные корни исходного уравнения являются также корнями многочлена  $y(x) = a_0x^5 + a_1x^4 + a_2x^3 + a_3x^2 + a_4x + a_5$ .

Постройте графики исходной функции и многочлена и установите диапазон аргумента, в котором эти функции совпадают.

Представьте обе функции в табличном виде в диапазоне их совпадения, решив задачу табулирования функций.

Определите погрешности полиномиальной функции.

## Задача 11.17

Функции  $y_1(x)$  и  $y_2(x)$  представлены в виде табл. 11.7.

**Таблица 11.7.** Табличное представление двух функций одного аргумента

$x$	1	2	3	4	5	6	7	8	9	10
$y_1$	1,2	3,7	5,2	8	9,1	10,8	11,5	13,2	15,1	16,8
$y_2$	23,4	17,2	15	13,4	10,2	7,1	4,3	1,2	0,5	0,1

Необходимо:

- определить путем интерполяции функций  $y_1(x)$  и  $y_2(x)$  математические модели  $\varphi_1(x)$  и  $\varphi_2(x)$ ;
- представить в виде таблицы функцию  $\varphi(x) = \varphi_1(x) + \varphi_2(x)$  путем табулирования;
- найти  $y_1 + y_2$  по данным таблицы и сравнить с данными табулирования;
- объяснить полученный результат.

## Задача 11.18

Даны две функции:  $y_1 = 2^x - 4x + 1$ ,  $y_2 = 2,3 \cdot 1,3^{0,65x}$ . Корни уравнений  $y_1(x) = 0$  и  $y_2(x) = 0$  имеют следующие значения:

$$x_1 = 3,8467; x_2 = 0,6394 \text{ уравнения } y_1(x) = 0;$$

$$x_3 = 2,46; x_{4,5} = 2,46 \pm 36,84j \text{ уравнения } y_2(x) = 0.$$

С помощью системы системы Derive 5 найти:

- многочлен  $f(x)$ , корни которого будут равны корням исходных уравнений;
- вторую производную функции  $f(x)$ ;
- интеграл  $\int_1^5 f(x)dx$ ;
- представить функции в виде таблиц и графиков.

## Задача 11.19

Дана нелинейная функция  $f(x) = 2e^{-0,8x} \sin x$ .

Необходимо:

- решить задачу аппроксимации Паде при  $x_0 = 0$ ;  $m = 2$ ;  $n = 3$ ;

- ☐ представить функции  $f(x)$  и  $\varphi(x)$  в виде таблиц и графиков;
- ☐ определить погрешность аппроксимации;
- ☐ найти производные  $f'(x)$  и  $\varphi'(x)$ ;
- ☐ вычислить интегралы:  $\int f(x)dx$  и  $\int \varphi(x)dx$ ,  $\int_0^1 f(x)dx$   
и  $\int_0^1 \varphi(x)dx$

Ответы:

- ☐ диапазон  $x$ , при котором обеспечивается точность аппроксимации три знака после запятой, равен 0—0,8;
- ☐  $\int_0^1 f(x)dx = 0,27728$ ,  $\int_0^1 \varphi(x)dx = 0,276698$ .

## Задача 11.20

Функция  $f(x)$  задана в виде табл. 11.8

**Таблица 11.8.** Табличное задание нелинейной функции

$x$	1	2	3	4	5	6	7	8
$y$	13,44	11,32	9,5	7,85	6,27	4,74	3,22	1,7

Известно, что функция интерполяции имеет вид:  
 $y(x) = ae^{-bx} + cx + 3.7$ .

Необходимо:

- ☐ найти функцию интерполяции, определить коэффициенты  $a, b, c$  (решение выполнить методом интерполяции точным в узлах);
- ☐ вычислить погрешность интерполяции в диапазоне значений  $x$  от 0,5 до 5 с шагом 0,5.