

СПРАВОЧНИК ПО СИСТЕМЕ СИМВОЛЬНОЙ МАТЕМАТИКИ DERIVE

М.: "СК Пресс", 1998. - 256 с., ил.

Описана 32-разрядная версия системы символьной математики (компьютерной алгебры) Derive 3.11 XM под MS-DOS и Derive 4.02 под Windows 3.1/3.11 и Windows 95. DERIVE обеспечивает проведение как простых, так и умеренно сложных вычислений в символьном (аналитическом) и в численном виде, имеет развитые возможности двумерной и трехмерной графики и богатые библиотеки внешних расширений. Приведено множество практических примеров применения системы для решения математических, научно-технических, статистических и экономических задач. Показаны возможности объединения DERIVE с текстовыми процессорами класса Word. Требуется небольших ресурсов ПК и отличается высокой скоростью и надежностью решения математических задач различной сложности.

Для широкого круга специалистов в области математики, физики и прикладных наук, а также для школьников, студентов высших учебных заведений и пользователей ПК.

Оглавление

Предисловие	9
Глава 1. Работа с математической системой Derive 3.1 1 под MS-DOS	
1.1. Общая характеристика системы Derive и ее развитие	13
1.1.1. Место Derive в семье компьютерных математических систем	13
1.1.2. Возможности системы Derive	16
1.1.3. Derive в образовании	19
1.2. Установка и загрузка системы	21
1.3. Главное меню системы Derive XM	22
1.4. Редактирование и ввод выражений (Autor)	24
1.4.1. Ввод выражений	24
1.4.2. Клавиши редактирования и управления курсором	25
1.4.3. Выделение выражений и подвыражений	26
1.4.4. Примеры ввода и редактирования выражений	27
1.4.5. Ввод греческих букв и проблема русификации Derive XM	28
1.4.6. Примеры работы с системой Derive XM	30
1.5. Построение математических выражений (Build)	33
1.6. Специальные вычисления (Calculus)	35
1.6.1. Вычисление производных	35
1.6.2. Интегрирование выражений	37
1.6.3. Вычисление пределов функций	40
1.6.4. Вычисление произведения членов ряда	41
1.6.5. Вычисление суммы членов ряда	42
1.6.6. Разложение функции в ряд Тейлора	43
1.7. Декларация новых определений (Declare)	44
1.7.1. Декларация функций	44
1.7.2. Декларация переменных	44

1.7.3. Декларация матриц и векторов	45
1.8. Раскрытие выражений (Expand)	47
1.8.1. Техника применения команды Expand	47
1.8.2. Синтез полинома по его действительным и комплексным корням	48
1.9. Факторизация (Factor)	49
1.9.1. Разложение чисел на простые множители	49
1.9.2. Разложение на части выражений	49
1.10. Работа с помощью (Help)	50
1.11. Перемещение по строкам документа (Jump)	51
1.12. Решение уравнений (soLve)	52
1.12.1. Решение нелинейных уравнений и неравенств	52
1.12.2. Вычисление действительных и комплексных корней полиномов	52
1.12.3. Решение неравенств и систем линейных уравнений	53
1.13. Управление вычислениями и подстановки (Manage)	53
1.13.1. Управление вычислениями	53
1.13.2. Подстановки	54
1.14. Задание глобальных опций (Options)	55
1.15. Выход из системы (Quit)	57
1.16. Работа с графикой (Plot)	58
1.16.1. Построение двумерных графиков в декартовой системе координат	58
1.16.2. Построение графика параметрически заданных функций	61
1.16.3. Построение графика функций в полярной системе координат	61
1.16.4. Построение графиков функций в трехмерном пространстве	62
1.17. Уничтожение строк (Remove)	66
1.18. Упрощение выражений (Simplify)	66
1.19. Работа с внешними устройствами (Transfer)	67
1.19.1. Работа с дисковыми накопителями	68
1.19.2. Работа с принтером	70
1.20. Перемещение строк (moVe)	71
1.21. Работа с окнами (Window)	71
1.21.1. Разбиение окон	71
1.21.2. Управление окнами	72
1.21.3. Примеры работы с окнами	72
1.22. Команда численных вычислений (approX)	75
1.23. Сообщения об ошибках	76
Глава 2. Язык программирования системы Derive	
2.1. Алфавит системы и комментарии	78
2.1.1. Понятие о входном языке Derive	78
2.1.2. Алфавит Derive	79
2.1.3. Проблемы русификации	79
2.1.4. Текстовые комментарии	79

2.1.5. Константы	80
2.2. Переменные и функции пользователя	80
2.2.1. Переменные и их определение	80
2.2.2. Функции пользователя и их декларация	81
2.2.3. Примеры применения объявленных переменных и функций	82
2.3. Арифметические и логические операторы и функции	83
2.3.1. Форматы чисел	83
2.3.2. Арифметические операторы	84
2.3.3. Логические операторы	85
2.3.4. Функция IF для создания условных выражений	86
2.4. Математические функции	87
2.4.1. Числовые функции	87
2.4.2. Кусочно- непрерывные функции	88
2.4.3. Функция генерации случайных чисел	88
2.4.4. Степенные (экспоненциальные) функции	90
2.4.5. Логарифмические функции	90
2.4.6. Тригонометрические функции	91
2.4.7. Обратные тригонометрические функции	92
2.4.8. Гиперболические функции	92
2.4.9. Обратные гиперболические функции	93
2.4.10. Примеры операций с элементарными функциями	93
2.4.11. Функции комплексного аргумента	94
2.4.12. Факториальные и комбинаторные функции	95
2.5. Функции математического анализа	96
2.5.1. Функции для вычисления пределов	96
2.5.2. Функции дифференцирования	97
2.5.3. Функция разложения в ряд Тейлора	98
2.5.4. Функции интегрирования	98
2.5.5. Функции суммирования и произведения членов рядов	98
2.5.6. Функции решения уравнений и неравенств и систем с ними	98
2.5.7. Функция выполнения итераций (ITERATES)	99
2.5.8. Примеры реализации итерационных вычислений	100
2.6. Векторные и матричные функции	101
2.6.1. Векторы и матрицы	101
2.6.2. Векторные функции и операторы	102
2.6.3. Матричные функции и операторы	104
2.6.4. Матричные операции в символьной форме	107
2.7. Статистические и финансовые функции	109
2.7.1. Функции ошибок	109
2.7.2. Основные статистические функции	110
2.7.3. Финансово-экономические функции	111
2.8. Реализация метода наименьших квадратов	113
2.8.1 Функция FIT	113
2.8.2. Линейная регрессия	113

2.8.3. Логарифмическая регрессия	114
2.8.4. Полиномиальная регрессия	115
2.8.5. Линейная регрессия общего вида	115
2.8.6. Регрессия для функции нескольких переменных	116
2.8.7. Нелинейная регрессия с выводом данных в графической форме	116
2.9. Функции градиентного и векторного анализа	118
2.10. Примеры и специальные приемы программирования	119
2.10.1. Пример вычисления модульной теплоемкости по Дебаю	119
2.10.2. Функции с параметрами-функциями	120
2.10.3. Задание функции для вычисления интеграла по формуле Уэддля	121
2.10.4. Минимизация функций	122
2.10.5. Организация циклов и условных выражений	123
2.10.6. Вычисление коэффициентов Берга	124
2.10.7. Функции выделения различных частей отношения	125
2.10.8. Несколько слов о рекурсии	125
2.11. Заключительные замечания по программированию	126
Глава 3. Описание библиотек внешних расширений	
3.1. Дополнительные векторные и матричные функции	127
3.2. Решение нелинейных систем уравнений	131
3.3. Решение дифференциальных уравнений	133
3.3.1. Частное решение для начальных условий $y = y_0$ при $x = x_0$	134
3.3.2. Общее решение, содержащее константу c	134
3.3.3. Обыкновенные дифференциальные уравнения первого порядка (развитые методы)	135
3.3.4. Примеры решения дифференциальных уравнений первого порядка	136
3.3.5. Обыкновенные дифференциальные уравнения второго порядка	137
3.3.6. Решение уравнения вида $y' = r(x)$ с начальным условием $y(x_0) = y_0$	138
3.3.7. Функция Рунге—Кутта	138
3.4. Решение рекуррентных уравнений	140
3.5. Дополнительные функции дифференцирования	141
3.5.1. Функции численного дифференцирования	141
3.5.2. Применение операций дифференцирования	142
3.6. Дополнительные функции интегрирования	144
3.7. Вычисление специальных математических функций	148
3.7.1. Интегральные показательные функции	148
3.7.2. Дополнительные вероятностные функции	149
3.7.3. Интегралы Френеля	151
3.7.4. Функции Бесселя и Эйри	152
3.7.5. Гипергеометрические функции	153
3.7.6. Эллиптические интегралы	154
3.7.7. Ортогональные полиномы	154

3.7.8. Дзета-функция, полилогарифм и дилогарифм (файл zeta.mth)	155
3.8. Графические функции	156
3.9. Паде-аппроксимация	159
3.10. Прочие функции	161
3.11. Описание вспомогательных файлов	163
Глава 4. Математическая система Derive 4.02 под Windows	
4.1. Пятиминутное знакомство с Derive под Windows	166
4.1.1. Запуск Derive под Windows	166
4.1.2. Правила работы с мышью	166
4.1.3. Начало работы	167
4.1.4. Графическая визуализация вычислений	169
4.1.5. Сохранение сессии на диске	172
4.2. Основы пользовательского интерфейса Derive под Windows	173
4.2.1. Главное меню	173
4.2.2. Панель инструментов и кнопки быстрого управления	174
4.2.3. Строка состояния системы	175
4.3. Работа с файлами (File)	175
4.3.1. Создание нового окна выражений (New)	176
4.3.2. Вызов диалогового окна загрузки файла документа (Open...)	176
4.3.3. Закрытие активного окна (Close)	177
4.3.4. Работа с подменю загрузки различных файлов (Load)	177
4.3.5. Запись документа с текущим именем (Save)	181
4.3.6. Вызов диалогового окна записи файла с заданным именем (Save As...)	181
4.3.7. Запись в форматах языков программирования (Write To)	182
4.3.8. Изменение текущего каталога с файлами (Change Directory...)	182
4.3.9. Вызов диалогового окна начала печати (Print...)	183
4.3.10. Предварительный просмотр страниц печати (Print Preview)	184
4.3.11. Установка параметров страницы печати (Page Setup...)	184
4.3.12. Завершение работы с системой Derive и выход из нее (Exit)	184
4.4. Редактирование выражений и документов (Edit)	185
4.4.1. Вывод окна редактирования выделенного выражения (Expression...)	186
4.4.2. Ввод аннотации для выделенного выражения (Annotation...)	187
4.4.3. Переход к заданному выражению (Go to Expression...)	187
4.4.4. Стирание строк с выражениями (Remove...)	188
4.4.5. Восстановление стертых строк (Unremove...)	188
4.4.6. Перемещение выражений (Move...)	188
4.4.7. Перенумерация строк (Renumber)	189
4.4.8. Копирование выражений в буфер обмена (Copy Expressions)	190
4.4.9. Выделение и копирование (Mark and Copy...)	190
4.5. Создание рабочего места математика на базе Derive и текстового процессора класса Word	191
4.6. Ввод данных (Author)	194

4.6.1. Ввод математических выражений (Expression...)	194
4.6.2. Задание вектора определенных размеров (Vector...)	194
4.6.3. Задание матрицы заданных размеров (Matrix...)	195
4.7. Преобразования выражений (Simplify)	197
4.7.1. Общее упрощение выражений (Basic...)	197
4.7.2. Расширение выражений (Expand...)	199
4.7.3. Факторизация выражений (Factor...)	200
4.7.4. Вычисления выражений в численном виде (Approximate...)	202
4.7.5. Подстановки (Substitute for)	202
4.8. Решение уравнений и неравенств (Solve)	204
4.8.1. Решение в символьном виде (Algebraically...)	204
4.8.2. Решение в численном виде (Numerically...)	205
4.8.3. Решение систем уравнений (System...)	207
4.9. Основные виды вычислений (Calculus)	208
4.9.1. Вычисление пределов функций (Limit...)	208
4.9.2. Вычисление производных функций (Differentiate...)	209
4.9.3. Разложение функций в ряд Тейлора (Taylor series...)	210
4.9.4. Вычисление неопределенных и определенных интегралов (Integrate...)	211
4.9.5. Вычисление сумм рядов (Sum...)	214
4.9.6. Вычисление произведений рядов (Product...)	215
4.9.7. Представление ряда в виде вектора (Vector...)	216
4.10. Декларация переменных и функций (Declare)	217
4.10.1. Декларация переменной и присваивание ей значений (Variable Value...)	217
4.10.2. Декларация области определения переменной (Variable Domain...)	217
4.10.3. Задание функции пользователя (Function Definition...)	218
4.10.4. Подменю управления форматом ввода/вывода (Algebra State)	219
4.11. Установка опций (Options)	221
4.11.1. Установка опций цвета (Color)	221
4.11.2. Установка опций печати (Printing)	222
4.11.3. Включение/выключение диалога (The Entry Dialog)	223
4.12. Задание и размещение окон (Window)	224
4.12.1. Создание дополнительного окна с выражениями (New Algebra View)	224
4.12.2. Создание нового окна для двумерной графики (New 2D-plot Window)	225
4.12.3. Панель инструментов окна двумерной графики	226
4.12.4. Главное меню окна двумерной графики	228
4.12.5. Графики параметрически заданных функций	230
4.12.6. Построение графиков функций в полярной системе координат	231
4.12.7. Работа с графическим курсором	231
4.12.8. Создание нового окна для трехмерной графики (New 3D-plot)	232

Window)	
4.12.9. Панель инструментов окна графиков (3D-plot)	233
4.12.10. Главное меню окна трехмерной графики	235
4.12.11. Каскадное расположение окон (Cascade)	236
4.12.12. Расположение окон по горизонтали (Tile Horizontally)	237
4.12.13. Расположение окон по вертикали (Tile Vertically)	237
4.12.14. Приведение в порядок расположения бирок (Arrange Icons)	237
4.12.15. Ввод или устранение инструментальной панели (Toolbar)	238
4.12.16. Ввод или устранение панели состояния (Status Bar)	239
4.12.17. Графика как средство визуализации математических понятий	240
4.13. Вызов справки о системе (Help)	243
4.13.1. Вызов контекстной справки (Contents)	243
4.13.2. Вызов индексного каталога (Index)	243
4.13.3. Вызов справки о самой справочной системе (Using Help)	246
4.13.4. Краткая информация о системе (About Derive)	246
Заключение	247
Приложение 1. Перевод сообщений Derive	248
Приложение 2. Интерпретация записи примеров для Derive 3.11	252
Литература	253

Предисловие

Основной движущей силой при разработке программы DERIVE было желание сделать работу с математикой более увлекательной и приятной.

**Алберт Рич, Джоан Рич, Дэвид Стаутмайер
(Создатели Derive)**

Компьютеры, от микрокалькуляторов [1] до ЭВМ большой производительности, изначально создавались для выполнения трудоемких математических расчетов. Для этого готовились специальные программы на различных языках программирования, например Фортране, Алголе, Бейсике, Форте или Паскале (см., например, [2, 3]). И по сей день такие сферы применения, как обработка текстовой и графической информации, работа с базами данных и электронными таблицами, остаются, в сущности, некоторым практическим приложением самых тривиальных математических методов. Недаром бытовым синонимом программного обеспечения компьютеров стал термин “математика компьютеров”.

Однако высочайшим достижением истинной компьютерной математики и компьютерного “интеллекта” стали системы компьютерной алгебры, выполняющие наряду с традиционными численными расчетами символьные (аналитические) вычисления. К сожалению, эти мощные и “широко развитые” экспертные системы были известны лишь крайне узкому кругу специалистов. Большинство же пользователей считало такие системы полезными лишь для профессионалов.

Тем временем персональные компьютеры решительно проникли в наши дома [4]. Вначале главной сферой их применения были игры и прочие мультимедийные приложения [5]. Но сейчас основная масса пользователей ПК (а их счет идет уже на миллионы) начала осознавать, что современный ПК — это не только усовершенствованная пишущая машинка и игровой автомат, а прежде всего гибкое и мощное вычислительное устройство, способное реально помочь в повседневной работе и учебе. Так, с помощью компьютера оказалось возможным обучение языкам, физике, химии и даже математике.

Как же помочь пользователям в обучении современным основам символьных (аналитических) вычислений и, главное, в их повседневном применении в работе? Эти задачи сейчас решает компьютерная алгебра — новейшее направление в информационных технологиях, получившее бурное развитие в 90-е годы именно благодаря появлению ПК. За последние несколько лет автор настоящей книги предсказывал в целом ряде своих обзоров и монографий, что такие системы для ПК должны занять одно из первых мест в арсенале не только математика, но и инженера, научного работника, студента и даже школьника [6–14, 17, 19, 21–23]. Так оно и случилось!

Уже появился и используется ряд систем компьютерной алгебры на массовых ПК: muMath, Reduce, MathCAD 3.0–7.0 [11–17], Derive [18–21], Maple V [22, 23], Mathematica 2/3 и др. [24–27]. Еще совсем недавно наши читатели почти не были знакомы с ними, ибо книги о конкретных математических системах такого класса у нас вообще не выпускались, а издательства (явно себе во вред) считали подобную литературу не слишком прибыльной. Первая книга автора о самой простой и массовой системе символьной математики DERIVE [18] была сдана в издательство еще в конце 1991 г., но вышла в свет лишь в 1996 г.

Однако сейчас прорыв символьной математики в образование стал очевидным. Следом за книгами автора и его многочисленными обзорами появились книги по крупным математическим системам MathCAD PLUS 6.0 [15, 16], Maple V [23] и Mathematica 2 [27].

Но выпуск одной-двух книг по каждой системе далеко не отражает степени практического применения систем и явно отстает от разработок их новых версий. Так, в руководстве пользователя по DERIVE [21] указано 33 книги по этой системе, изданные за рубежом только к 1995 г. А по другим указанным выше системам еще до выхода в свет книг [23, 27] появились новые версии, например MathCAD 7.0, Maple V R4, Mathematica 3 и др. После издания книги [18] вышло уже несколько новых версий DERIVE, в том числе новейшая 32-разрядная версия 4.02 под Windows 3.1/3.11 и Windows 95. Таким образом, мы вновь оказались в начале витка развития математических систем — теперь уже символьной математики — и вновь в условиях острого дефицита необходимой литературы.

Эта книга хотя бы частично заполняет существующий пробел, что особенно важно в связи с наметившимся ростом интереса к образованию. Такая тенденция вполне понятна: все наконец-то поняли, что в условиях свертывания неконкурентоспособного производства образование позволяет куда легче адаптироваться к новым условиям жизни.

Здесь впервые в России (и в странах СНГ) описываются две новейшие версии одной из самых удачных систем компьютерной алгебры — Derive 3.11 под MS-DOS и Derive 4.02 под Windows 3.1/3.11/95. Обе системы совместимы, имеют практически одинаковый набор функций и команд и могут обмениваться своими файлами. Но они рассчитаны на два класса операционных систем: на все еще хорохорящуюся старушку MS-DOS и на более современные операционные системы класса Windows, включая новейшую Windows 95.

Книга может служить доступным введением в практику работы с системами компьютерной алгебры. Она наверняка даст толчок к внедрению этих систем в наше образование, научное творчество и в инженерное проектирование — словом, во все области, требующие применения самых современных математических методов. Derive a Mathematical Assystent (математический помощник DERIVE), как полностью называется эта система фирмы Soft Warehous (США), сочетает обширные возможности численных и символьных вычислений с весьма неплохой графикой. Она, пожалуй, является лучшей массовой системой для начального знакомства с символьными (аналитическими) математическими компьютерными системами.

DERIVE — самая маленькая из систем компьютерной алгебры. Образно говоря, это подлинная жемчужина символьной математики. Авторы некоторых книг тенденциозно пишут о больших математических системах MathCAD PLUS 6.0/7.0 или Maple V R3/R4 как о предназначенных “для всех”, не смущаясь даже тем, что сами разработчики явно указывают на их профессиональное назначение и что аппаратные ресурсы для использования таких систем доступны далеко не всем нашим владельцам ПК.

Между тем именно DERIVE поистине система для всех! Даже последняя версия Derive 3.11 под MS-DOS может устанавливаться на “древних” ПК класса IBM PC XT без жесткого диска. Затраты на передачу такой системы по Internet или по локальной сети весьма невелики. При этом система имеет многооконный интерфейс пользователя и управляется простой (хотя и не самой современной) системой меню. Она открывает доступ к огромным возможностям систем символьной математики даже тем пользователям, которые по сей день вынуждены работать с ПК классов 386 и 486.

Система Derive 4.02, обладая удобным (и в тоже время удивительно простым) интерфейсом приложений под Windows 95, занимает на жестком диске менее 8 Мбайт памяти, что в несколько раз меньше, чем “пожирают” MathCAD 6.0, Mathematica 2 и Maple V R3. А что говорить об их новейших версиях, имеющих явную тенденцию к непомерному разбуханию — Mathematica 3, к примеру, занимает на жестком диске уже около 120 Мбайт, а инсталляционная версия MathCAD 7.0 не помещается в сотне мегабайтов.

Между тем новые 32-разрядные версии DERIVE могут обращаться к расширенной памяти до 4 Гбайт, что обеспечивает построение весьма развитых деревьев решений и позволяет справляться со сложнейшими задачами символьной математики. Однако эти версии DERIVE могут прекрасно работать и с памятью (ОЗУ) всего в 2 Мбайта.

Русская поговорка гласит: “Мал золотник, да дорог”! Это вполне относится к системе Derive 3.11. Благодаря ее реализации на языке искусственного интеллекта *muLISP* ХМ эта новая система, размещенная на дискете 1,2 Мбайта, допускает гораздо меньшую вероятность ошибок в символьных вычислениях, чем мировые лидеры среди систем символьной математики (*Mathematica 2* и *Maple V*), “съедающие” мегабайты памяти в ОЗУ и десятки мегабайтов на жестком диске и вызывающие у нормального пользователя не столько уважение, сколько страх перед своей сложностью и тысячами функций. Между тем, освоив DERIVE, пользователи начисто лишаются такого страха и могут (если имеют доступ к современному ПК) начать освоение этих мощных профессиональных математических систем.

Эта книга является справочным руководством и руководством пользователя по версиям Derive 3.11 под MS-DOS и 4.02 под Windows. Она не повторяет фирменного описания этих систем и содержит множество авторских примеров их применения. Математические алгоритмы и методы вычислений в книге описаны очень кратко, их детали заинтересованный читатель может найти в специальной литературе [28–32]. Первая часть книги (по версии 3.11) — это существенно переработанная и дополненная книга автора [18], а вторая — описание версии 4.02 под Windows, которое в нашей литературе дается впервые (о ней нет даже статей). Книга составлена таким образом, что ее могут использовать обладатели любой версии DERIVE, даже 2.0—2.6.

Автор обращает особое внимание читателей на то, что книга является целым справочным пособием. Пользователям MS-DOS версий DERIVE полезно ознакомиться с главой 4, поскольку в ней можно найти множество дополнительных примеров применения DERIVE. Они вынесены в эту главу, потому что именно DERIVE под Windows позволяет наиболее наглядно представить возможности этой системы, особенно в части визуализации процессов поиска решений математических задач и полученных результатов. А пользователям последней версии Derive 4.02 под Windows необходим материал глав 2 и 3, содержащих описание основополагающих концепций систем класса DERIVE, принципов их применения, программирования и расширения с помощью внешних библиотек. Тем не менее для начального знакомства с новой версией 4.02 под Windows достаточно прочесть главу 4 отдельно.

Благодарности

Книга подготовлена по результатам инициативных работ Смоленского областного центра новых информационных технологий в образовании (СОЦНИТ) и кафедры физической и информационной электроники Смоленского государственного педагогического института.

Автор выражает глубокую признательность С. В. Бирюкову, представителю фирмы МНИИТЦ “Скан”, за плодотворное сотрудничество и своевременно предоставленную информацию по различным версиям системы DERIVE. Автор признателен и Л. Д. Лозинскому, который первым обратил его внимание на жемчужину символьной математики — систему DERIVE.

И, конечно, нельзя не высказать теплых слов в адрес фирмы Soft Warehouse и всех ее сотрудников, создавших такую замечательную систему, как Derive a Mathematical Assystent, и сумевших найти ей должное место на рынке современных математических систем.

Предупреждения и пожелания

Автор и издательство сделали все возможное, чтобы материал книги был доступен, актуален, объективен и точен. Однако компьютерная алгебра — новое направление, ее терминология и перевод отдельных терминов ее еще не устоялись, порою могут трактоваться двояко. Кроме того, в большинстве примеров для системы Derive 3.11 под MS-DOS математические выражения представлены в текстовом виде. В приложении 2 дается соответствие основных элементов таких выражений общепринятым математическим символам.

Будем признательны всем, кто выскажет свое мнение о книге. Отзывы и пожелания следует направлять в адрес издательства или по адресу: СОЦНИТ, 214013, Смоленск, Энергетический пр., 1, СФМЭИ, СОЦНИТ, автору книги.

Об авторе

Доктор технических наук профессор Владимир Павлович Дьяконов (рожд. 1940 г.) — известный специалист в области электроники и информатики. Он автор около двух десятков книг, свыше 400 научных и учебно-методических статей и более 60 изобретений. Профессор В. П. Дьяконов — основатель и директор Смоленского областного центра новых информационных технологий в обучении, заведующий кафедрой физической и информационной электроники Смоленского государственного педагогического института.

Владимир Павлович первым всерьез заговорил о новом направлении компьютерной математики — математических системах для численных и аналитических вычислений на малых вычислительных машинах. Он впервые в бывшем СССР опубликовал общедоступные справочники по расчетам на микрокалькуляторах и персональных ЭВМ, а затем и по математическим системам Eureka, MathLAB, MathCAD и Derive (ранние версии под MS-DOS). Тираж только этих книг достиг полутора миллиона экземпляров!

Будучи специалистом широкого профиля, В. Дьяконов опубликовал монографии по электронным схемам на лавинных и мощных полевых транзисторах и первую в России обширную энциклопедию по мультимедиа (Москва, изд-во АБФ, 1996). Известна и одна из последних книг В. П. Дьяконова для широкого читателя “Компьютер в быту”, изданная крупным издательством “Русич” (Смоленск).

Написанная совместно с коллегами по Смоленскому педагогическому институту крупная монография об основах программирования на языке QBasic — *The Revolutionary Guide to QBasic* — издана в Великобритании на английском языке (Wrox Press Ltd., UK, 1996). Владимир Павлович автор ряда книг о языках программирования Бейсик, Лого, Форт и Паскаль и об их применении для математических расчетов. Специалистам известно множество его обзорных и научно-популярных статей в компьютерных журналах “Мир ПК”, “Монитор-Аспект”, “PC Week/RE” и др.

Книга, которую вы держите в руках, собственно говоря, второе издание справочника по одной из самых интересных и массовых систем компьютерной алгебры — Derive. Здесь рассмотрены новейшие 32-разрядные версии этой системы, работающие и в MS-DOS, и в Windows 3.1/3.11/95. Как и другие справочники проф. В. П. Дьяконова, эта книга написана прекрасным языком и совершенно полно раскрывает тему. Она вполне может служить практическим руководством пользователя Derive.

Глава 1. Работа с математической системой Derive 3.11 под MS-DOS

1.1. Общая характеристика системы и ее развитие

1.1.1. Место Derive в семье компьютерных математических систем

Роль математики в науке, технике и образовании исключительно велика. Без математических расчетов или их результатов не обходится серьезное проектирование ни одного промышленного изделия. Большое внимание уделяется изучению математики в школах, техникумах и тем более в высших учебных заведениях. Многие миллионы наших сограждан, кто по влечению, а кто поневоле и даже по принуждению, ежегодно осваивают и применяют современные математические методы. Однако их эффективное применение на практике требует больших затрат времени и определенных способностей. Именно стремление облегчить математические расчеты привело к изобретению и развитию компьютеров.

И все же десятилетиями компьютеры помогали в решении математических задач лишь узкой группе специалистов, занятой сложными и трудоемкими расчетами. При этом компьютеры скорее не облегчали, а усложняли решение многих повседневных математических задач. Ведь пользователь был вынужден не только математически и алгоритмически грамотно поставить задачу, но и освоить работу на самом компьютере, изучить основы информатики и довольно сложные методы численных расчетов, освоить один или несколько языков программирования, составить программы расчетов и провести их трудоемкое и ответственное тестирование. Без всего этого программы пользователя были не более чем кустарной подделкой дилетантского уровня, обреченной лишь на захламление архивов создателей таких программ. Главным достоинством компьютеров была возможность очень быстрого выполнения математических вычислений.

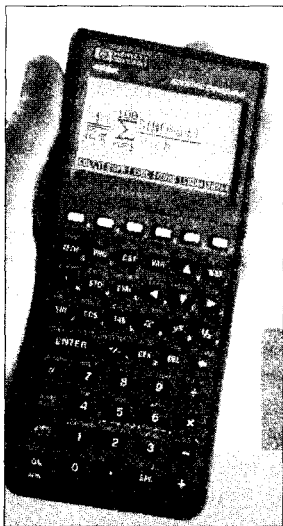
С появлением программируемых микрокалькуляторов и персональных компьютеров (ПК) вычислительные машины стали доступны рядовому инженеру или студенту [1–3]. Однако подход к их применению оставался прежним. И лишь в последние годы наметился существенный прогресс в разработке интегрированных математических систем, резко уменьшивших затраты времени на их освоение и программирование. Первые такие системы (например, Eureka фирмы Borland, MathCAD 2.0–2.5 фирмы MathSoft или MatLAB фирмы MathWork) реализовали возможность численных методов анализа и имели удобную для пользователя рабочую среду.

В то же время эти системы демонстрировали поразительную беспомощность при решении алгебраических задач в символьном (формульном) виде. Они закидывали пользователя сообщениями об ошибках при попытке вычислить в общем виде значения выражений вида $(a^2 - 2 * a * b - b^2) / (a - b)$ или $\sin(x) / \cos(x)$, хотя результаты вычисления этих выражений очевидны даже для не слишком успевающего ученика средней школы. Ну а вычисление интегралов или производных в символьном виде могло лишь привидеться в бреду!

Новое поколение интегрированных математических систем освоило компьютерную алгебру. Так называют совокупность средств, способных выполнять аналитические преобразования и вычисления, например вычислять неопределенные и определенные интегралы в символьном виде, подставлять одни выражения на места других, находить производные и пределы любых функций и т. д. и т. п.

Уже сейчас известно множество математических систем и пакетов, позволяющих автоматизировать проведение как числовых, так и символьных (формульных) вычислений. Появились даже программируемые микрокалькуляторы, способные на это (например, HP28С, HP48С или HP95LX фирмы Hewlett-Packard). Внешний вид калькулятора HP48С, выполняющего символьные операции и имеющего графические средства, показан на рис. 1.1.

Рис. 1.1
Внешний вид
научного
графического
калькулятора
HP48С
с возможностью
выполнения
символьных
операций



А калькулятор HP-95 даже обзавелся встроенной системой Derive, и это, пожалуй, лучший комплимент в ее адрес, благо такой крупный и почетный производитель измерительных приборов и ультрасовременных компьютеров, как фирма Hewlett-Packard, выбрала Derive для своего уникального калькулятора. Он, кстати, пользуется успехом и выпускается по сей день, хотя в моду вошло применение персональных компьютеров, в том числе самых портативных (меньших по размерам, чем Notebooks).

К сожалению, многие из таких систем для персональных компьютеров (MapleV R3/R4, Mathematica 2/3, MathCAD 6.0 PLUS и др.) пока очень громоздки и требуют от ПК больших аппаратных ресурсов. Их можно использовать лишь на дорогих и дефицитных ПК класса 486/Pentium с большим объемом ОЗУ — более 8 Мбайт. Но даже на таких ПК эти системы часто производят символьные операции в явно замедленном темпе, нервирующем пользователей, привыкших к чуть ли не мгновенному решению на ПК умеренно сложных численных задач.

Широкое применение математических систем все еще ограничено и иными факторами. Прежде всего надо указать на то, что никакие красочные меню в таких системах не освобождают пользователя от понимания сути математических

команд и методов, реализованных в системах. Без детального описания систем их применение под силу лишь очень квалифицированным пользователям и больше напоминает разгадку кроссвордов, чем реальную работу с системами.

Этим математические системы принципиально отличаются от других, интуитивно понятных программных систем, например текстовых или графических редакторов. Поэтому существенно возрастает роль литературы, посвященной описанию таких систем (тем более в условиях дефицита фирменных руководств по применению систем и их переводов на русский язык, как правило, отсутствующих на дискетах и CD-ROM).

От многих программных пакетов и систем автоматизации математических расчетов система Derive отличается тем, что удачно сочетает возможности проведения численных и символьных вычислений с простотой и требовательностью к используемой технике (может работать на ПК класса IBM PC XT с умеренными характеристиками, оснащенными лишь дисководами для гибких магнитных дисков).

Фактически это единственная малая математическая система, обеспечивающая решение задач компьютерной алгебры в диалоговом режиме с вполне приемлемым быстродействием и хорошими графическими возможностями на любых IBM-совместимых ПК, в том числе самых простых и доступных. Ее возможности (многоцветная графика, эффективность применения и скорость работы) намного возрастают при работе на современных ПК с микропроцессорами 486 и Pentium, имеющими повышенную производительность. А появление системы Derive 4.02 под Windows дает возможность ощутить все преимущества пользовательского интерфейса приложений под операционные системы Windows 3.1/3.11/95.

ЭВМ и программные системы, выполняющие символьные вычисления и способные выдавать результаты в виде аналитических зависимостей, известны давно. Лидирующая роль в разработке таких ЭВМ в СССР принадлежала школе академика В. М. Глушкова, создавшей для проведения символьных вычислений малые инженерные ЭВМ серии "Мир" с языком "Аналитик". К сожалению, это направление вычислительной техники в дальнейшем не было поддержано в должной мере, и лидерство перешло к иностранным разработчикам таких средств.

За рубежом был создан ряд языков программирования и программных систем для символьных операций: muMATH, REDUCE, Maple, Mathematica и др., заложивших основу для развития компьютерной алгебры [28, 29, 34]. Среди этих систем одной из самых простых и получивших массовое распространение была система muMATH, реализованная на многих мини- и микро-ЭВМ. Фирма Soft Warehouse разработала на основе этой системы в последние годы универсальную математическую систему Derive a Mathematical Assistant (далее просто Derive) [35–38], описанию которой и посвящена наша книга.

Derive в переводе означает "извлекать" или "наследовать". Видимо, оба этих значения учитывали разработчики системы. Derive унаследовала лучшие черты от самой массовой системы компьютерной алгебры — muMATH. Поэтому, несмотря на новизну, Derive тщательно опробованная, надежная и быстрая система. В своей обобщающей обзорной статье [38] главные создатели системы Derive отмечают, что на ее создание (версия 2.01) они затратили 10 лет.

Справедливо и то, что из множества математических формул и выражений, различаемых системой, и невообразимо большого числа их комбинаций даже опытный пользователь может извлечь для себя новые и порою неожи-

данные математические закономерности и результаты. В этом нет ничего удивительного — ядро Derive содержит около 1000 функций, написанных на языке MuLISP, и 23 тысячи программных строк. Для сравнения можно отметить, что громадная система Mathematica 2.2.2 содержит в ядре, написанном на языке Си++, столько же функций. Удивительная компактность ядра Derive связана именно с языком искусственного интеллекта MuLISP, прекрасно подходящим для реализации сложных рекурсивных функций, взаимодействующих друг с другом в ходе осуществления символьных преобразований. О языке LISP можно прочитать в [33]. Это один из лучших языков для создания экспертных систем, к которым и относятся системы класса Derive.

Среди математических систем по-своему уникальна MathCAD [11–17]. Ее документы (т. е. программы, комментарии, результаты вычислений и графики) имеют естественный для научно-технической литературы вид. Однако при вводе математических выражений в этой системе пользователь должен постоянно помнить о правилах ввода специальных математических символов, которых нет на клавиатуре ПК класса IBM PC. Далеко не сразу запоминаются и правила редактирования формул.

В других системах, например MatLAB [10, 11], или в обычных языках программирования высокого уровня вместо математических знаков вводят слова, обозначающие математические функции и операторы. Например, sqrt или sq вместо знака квадратного корня (а иногда возведения в квадрат). Надо отметить, что большинство пользователей ПК настолько привыкло к таким обозначениям, что они не только не вызывают особого протеста, но и кажутся предпочтительными перед бесконечным тыканьем курсором мыши в многочисленные панели с математическими знаками (что необходимо сделать при наборе формул в MathCAD или в текстовых процессорах класса Word).

Derive и здесь имеет существенное достоинство: ввод математических символов с клавиатуры выполняется как набором специальных математических символов из панели, так и вводом (в версии под Windows) соответствующих слов. Выбирайте, что нравится! Однако на экране дисплея эти слова порождают изображение соответствующего математического символа. Таким образом, Derive объединяет достоинства привычной символики математических выражений с правилами их ввода, характерными для разных систем и языков программирования (например, Basic или Pascal). Особенно удачно и, главное, очень просто пользовательский интерфейс Derive реализован в версии 4.02 под Windows.

1.1.2. Возможности системы Derive

Derive — мастерица на все руки. Вот далеко не полный перечень задач, которые способна решать система даже без внешних расширений (библиотек):

- арифметические и логические операции, вычисление алгебраических, тригонометрических, гиперболических, обратных тригонометрических, обратных гиперболических, статистических и финансово-экономических функций (расширенный набор), ряда специальных математических функций;
- действия над числами произвольной разрядности и при различной системе счислений (основание чисел от 2 до 36);
- операции с действительными и комплексными числами, представление их в дробно-рациональной форме;

- символьные операции с полиномами, дробно-рациональными функциями, функциями одной и многих переменных, включая разложение полиномов на простые множители, вычисление их действительных и комплексных корней, нахождение числовых значений и пр.;
- символьное и численное дифференцирование и интегрирование, вычисление сумм и произведений элементов рядов, вычисление пределов функций, нахождение разложений в ряд Тейлора в окрестностях заданной точки;
- числовые и символьные операции с векторами и матрицами, элементы которых могут быть как числами, так и арифметическими выражениями;
- преобразования формул (различные подстановки, приведение в порядок по степеням заданных переменных, разложение на множители, упрощение и пр.);
- решение задач теории поля и векторного анализа, включая вычисление градиента, дивергенции и потенциала;
- построение двумерных и трехмерных графиков, графиков функций, заданных в параметрической форме, и графиков в полярной системе координат.

Средства графики Derive позволяют строить двумерные графики в полярной и декартовой системах координат и трехмерные графики с применением алгоритма удаления невидимых линий или без него. Графики автоматически масштабируются для получения их максимальных (при заданном типе дисплея) размеров. Возможно создание на экране дисплея многих окон и вывод в них различной информации, как текстовой, так и графической. При использовании дисплеев EGA и VGA можно реализовать возможности цветной графики.

Таким образом, Derive является универсальной математической системой, ориентированной на решение весьма широкого круга математических и научно-технических задач. Тем не менее современные версии Derive — это расширяемые системы, способные легко адаптироваться под решение специальных задач пользователя. Они поставляются с развитой библиотекой функций, существенно расширяющей и без того немалые возможности системы. Пользователь может создавать и свои библиотеки, окунувшись в магию программирования задач компьютерной алгебры.

Правда, задача расширяемости системы решена здесь не столь изящно, как, скажем, в системе MatLAB, где любое внешнее расширение без специальной загрузки можно тут же использовать для расчетов наряду со встроенными средствами. Derive позволяет готовить расширения (библиотеки функций) и записывать их в виде файлов. Для использования необходимых расширений нужно предварительно загрузить их с магнитного диска, что подчас требует заметно большего времени, чем в системе MatLAB. Зато Derive может работать практически на любых типах ПК класса IBM PC (даже без жесткого диска и математического сопроцессора) и для размещения всей системы достаточно гибкого диска 3,5 дюйма с форматом 1,2 Мбайта (для версии 3.11). Столько же система занимает и на жестком диске.

С разработкой 32-разрядных версий Derive, общающихся с памятью до 4 Гбайт, эта система становится пригодной для решения математических задач практически любой сложности с использованием библиотек ее расширения больших размеров. К примеру, автор лишь один раз получил на своем ПК Pentium 100 с ОЗУ 16 Мбайт сообщение о нехватке памяти, и то это случилось при случайном неверном задании глубоко рекурсивной функции, обреченной на вечную работу.

Derive 3.11 может работать и при отсутствии на ПК математического сопроцессора. Система поддерживает основные типы видеоадаптеров: CGA, MCGA, EGA, VGA, SVGA, Hercules и др. Она позволяет в стилизованной форме выводить тексты документов на печать. При установке на ПК драйверов кириллицы возможно задание текстовых комментариев на русском языке и даже задание русскоязычных идентификаторов. А система Derive 4.02 может работать совместно с текстовыми редакторами WordPad и Word 6.0/7.0, что позволяет готовить удобные отчеты с русскоязычными текстами и вставками из сессий работы Derive. Кстати говоря, развитие всех математических систем сейчас идет по пути интеграции их с современными текстовыми процессорами, что пригодится для создания рабочих мест учащихся, инженеров и ученых.

Для расширения своих возможностей и подготовки библиотек функций Derive обладает входным языком программирования сверхвысокого уровня, манипулирующим с операторами и функциями. Он поразительно прост и не требует от пользователя больших затрат времени на освоение (даже Бейсик куда сложнее, чем этот язык). Описание входного языка Derive дано в главе 2. А в главе 3 описаны библиотеки внешних расширений Derive, которые послужат ярким примером применения языка программирования Derive для решения множества математических задач — как простых и достаточно очевидных, так и самых сложных, например таких, как решение в аналитическом виде систем дифференциальных уравнений.

Языком реализации Derive является язык muLISP — одна из реализаций известного языка высокого уровня LISP, ориентированного на решение задач искусственного интеллекта и построения “разумных” экспертных систем [33]. Именно использование языка muLISP придает системе Derive интеллектуальность, которой так не хватает малым математическим системам Eureka, MathCAD (до версии 3.0) или MatLAB. В статье [21] описаны особенности реализации системы Derive и языка muLISP.

Хотя основное назначение Derive — символьные вычисления, система обладает превосходными возможностями выполнения численных расчетов с практически произвольной разрядностью. Так, она способна вычислить число π с точностью до 500 000 знаков, используя известную формулу для вычисления этого числа:

$$4/\pi = \sum_{m=0}^{\infty} \frac{(-1)^m (1123+21460*m)(1*3*...*(2*m-1)*(1*3*...*(4*m-1))}{882^{2*m+1} * 32^m * (m!)^3}$$

Предшественник Derive 3.11 XM система Derive Professional XM на ПК 486SX (с частотой работы микропроцессора 25 МГц и ОЗУ в 3 Мбайта) “все-го” за неделю работы вычислила по этой формуле указанное число точных цифр числа π ! Последние 20 цифр следующие: 4042487602513819524. Как говорится, не верите — проверьте!

Этот пример наглядно показывает, как успешно Derive может использоваться при решении сложнейших задач теории чисел, точных вычислениях с произвольной разрядностью и при реализации и проверке различных численных методов. Но особый интерес представляет ее применение для решения задач символьной математики — компьютерной алгебры.

1.1.3. Derive в образовании

Попытки обучить простых смертных аналитической математике в школах и вузах дают, мягко говоря, не слишком впечатляющие результаты. Даже студенты старших курсов вузов, фундаментально изучившие высшую математику, проявляют поразительную забывчивость при необходимости применения серьезных математических методов на практике. Речь, конечно, не идет об отдельных вундеркиндах, для которых математика стала смыслом их жизни и из которых в дальнейшем и выходят академики или хотя бы преподаватели математики в вузах и университетах. Речь об основной массе учащихся, которые создают космические корабли и ракеты, разрабатывают и производят бытовую сантехнику и торгуют на лотках пивом и Coca-Cola.

Derive вполне может сыграть роль палочки-выручалочки для большинства учащихся при обучении математике. В этой роли система широко известна за рубежом и интенсивно развивается. Она используется при преподавании математики во многих странах мира: уже к 1995 г. Derive применялась во всех школах Австрии, Словении, Южного Тироля (Италия), в 2500 школах Германии, в половине школ Португалии, была рекомендована для применения во Франции и т. д. Сейчас этот список намного шире.

За рубежом Derive описана во многих сотнях статей и в десятках книг. Создана группа пользователей этой системы, выпускающая журнал "Derive User Group Newsletter". С 1994 г. издается журнал "The International Derive Journal". Кроме того, организована электронная доска объявлений — (217) 337-0926 (США) и имеется адрес электронной почты (Internet) в Европе — cain@can.nl.

Серьезным достижением разработчиков Derive стал выпуск 32-разрядной профессиональной версии Derive XM Professional, ориентированной на ПК 386/486/Pentium. Эта версия работает с расширенной памятью (ОЗУ не менее 2 Мбайт) и практически исключает появление сообщений о нехватке памяти при проведении сложных символьных преобразований, чем иногда грешили обычные версии Derive. На ее основе и создана последняя версия системы под MS-DOS — Derive 3.11 XM.

Работа с Derive 3.11 XM практически ничем не отличается от работы с ранними версиями Derive. Различие лишь в том, что последние используют ОЗУ емкостью до 640 Кбайт, тогда как Derive XM использует всю память ПК — теоретически до 4 Гбайт. Если Derive ориентирована на решение простых и умеренно сложных задач, то Derive 3.11 XM может решать самые "крутые" задачи символьной математики с применением обширных библиотек расширений и глубоких рекурсий.

По сравнению с прежними версиями в Derive 3.11 возможны более естественное задание векторов и матриц, векторизация операций, упрощение логических выражений и построение таблиц их истинности, расширенные функции выделения частей выражений и итераций, сжатие изображения на экране, построение графиков неявно заданных функций, автоматический выбор масштаба построения графиков, улучшенные средства вывода надписей на графиках, печать на лазерных принтерах и т. д.

У нас внедрение в образование математических систем, особенно компьютерной алгебры, выглядит порою просто трагикомично. Преподаватели школ и вузов по-разному относятся к автоматизации святой святых — математических преобразований. Одни начисто отвергают возможность применения

таких систем в образовании, утверждая, что они якобы отучают учащихся от математических навыков и интуиции. Другие считают, что нечего учить школьников большинству в целом тривиальных математических преобразований вообще, коль их уже освоили компьютеры.

Нелепы и те и другие крайности. Вспомним, что не так уж давно педагоги-консерваторы отрицали полезность авторучек (портят почерк) и калькуляторов (отучают от устного счета). Сейчас эти доводы звучат столь же смешотворно, как отказ от поездки на поезде на том лишь основании, что пешком ходить полезнее.

Нет никаких серьезных причин отказываться от ПК в аналитических вычислениях и преобразованиях, коль большая часть из них формализована и подчиняется автоматизации. Более того, за рубежом появились полноценные учебники, в которых изучение математики базируется на постоянном общении школьников и студентов с системой Derive [35–37]. Их авторы наглядно показывают, что Derive — подлинная находка для педагога-новатора, эффективно помогающая в изучении математики.

Разумеется, постоянно надо помнить о том, что результаты символьных преобразований часто бывают неоднозначными и в зависимости от примененных правил могут приводить к разительно отличающимся результатам. Показать их идентичность подчас не менее сложно, чем выполнить сами преобразования. К тому же современный уровень развития систем компьютерной алгебры не исключает даже грубых ошибок в выполнении символьных вычислений или отказа от их выполнения. В таких случаях компьютеру нужна квалифицированная помощь: надо подсказать правильное направление преобразований. Но и тогда система Derive проявляет себя наилучшим образом: по мизерному количеству таких ошибок она обогнала куда более мощные системы Maple V и Mathematica 2 [34].

Вообще же правомерно рассматривать системы компьютерной алгебры как мощный инструмент в руках педагогов, учащихся, инженеров и ученых. Эффективность и методическая ценность такого инструмента всецело зависит от умения преподавателя или учащегося применять его на практике. Здесь та же ситуация, что с ножом: хирург с его помощью спасает людей, а бандит калечит и убивает.

В общем, при использовании системы Derive разумно придерживаться золотой середины. Тогда, автоматизируя рутинные, а подчас и довольно сложные математические расчеты и преобразования, система Derive вовсе не отвергает математическую интуицию пользователя и его творческое участие в их выполнении. Напротив, она помогает пользователю приобрести такую интуицию без порою бессмысленных и огромных затрат времени и частенько без слез, ой, как нередко сопровождающих принудительное изучение математики в школах и вузах. Derive позволяет быстро и ловко опробовать различные подходы к решению математических задач, которые из-за их трудоемкости нередко просто отвергаются.

Графика Derive примитивна, если сравнивать ее с графикой больших математических систем — MatLAB, MathCAD, Maple V или Mathematica 2/3. Тем не менее она вполне достаточна для такой важнейшей задачи, как визуализация вычислений. Между прочим, именно эту проблему Крейг Баретт, президент фирмы Intel, ведущей компании в мире по производству микропроцессоров, назвал одной из важнейших мегатенденций в развитии компьютеров XXI века.

1.2. Установка и загрузка системы

Derive 3.11 может работать как на ПК, оснащенном только одним-двумя накопителями на гибких магнитных дисках, так и на ПК с накопителем на жестких дисках. В первом случае для загрузки системы достаточно вставить диск с системой в накопитель A (или B) и дать команду:

```
A:\derive <ENTER>
```

где <ENTER> означает нажатие клавиши ENTER (в дальнейшем ее нажатие для фиксации той или иной команды особо отмечаться не будет).

Для ПК с жестким диском надо создать каталог с каким-либо именем, например DER или DERIVE, и скопировать в него все файлы с гибких дисков, на которых поставляется система, что происходит автоматически во время ее инсталляции. Для этой же цели обычно используются и возможности администратора диска Norton Commander. Пуск системы при этом выполняется просто указанием на выполняемый файл derive.exe.

Как отмечалось, Derive является расширяемой системой. Ее расширения можно загрузить сразу вместе с системой. Пусть, например, нужно расширить систему включением в ее состав библиотечного модуля `bessel.mth`, содержащего вычисления функций Бесселя, широко используемых в теории колебаний и фильтров. Для этого достаточно использовать команду:

```
a:\derive besel.mth
```

В процессе загрузки расширяемой системы текст расширений появится на экране дисплея, подтверждая дополнение системы. С помощью команды `Remove` можно очистить экран от этого текста (если это нужно), причем все расширения сохраняются в ОЗУ. Загрузить расширения можно и потом с помощью команды `Load`. Она имеет опцию `Utility`, позволяющую проводить загрузку отлаженных библиотек без вывода их текстовых документов на экран дисплея. При этом загрузка идет намного быстрее, чем в случае загрузки с выводом сообщений на экран дисплея.

Поскольку библиотечные файлы имеют текстовый формат, то заинтересованный пользователь может легко составить из имеющихся файлов свою библиотеку функций и дополнить ее определениями найденных функций. Не стоит включать в такую библиотеку слишком много функций — их должно быть не более нескольких десятков, хотя сама система воспринимает куда большее число определений новых функций и ограничения на их число в новых версиях практически сняты.

Система содержит также демонстрационные файлы с именами типа `*.dmo`. Они загружаются с помощью команды `Load` с опцией `Demo`. Демонстрационные файлы можно просмотреть, используя команду `Simplify`. В меню `USER` находится ряд расширений `Derive` и демонстрационных файлов, созданных пользователями, являющимися наиболее яркими поклонниками этой системы. Вы также можете стать фанатом, направив свои разработки в адрес фирмы `Soft Warehouse` (ее адреса и телефоны можно найти в справочном разделе системы `Derive`). Кроме того, в этом каталоге имеются текстовые файлы дополнительной документации.

Перед работой с `Derive` рекомендуется просмотреть два текстовых файла: `derive.doc` (краткая документация по системе) и `readme` (последние изменения

и сообщения). Просмотр этих файлов с помощью любого текстового редактора позволяет получить первое впечатление о возможностях системы и некоторых дополнительных данных, отсутствующих даже в фирменной документации.

1.3. Главное меню системы Derive XM

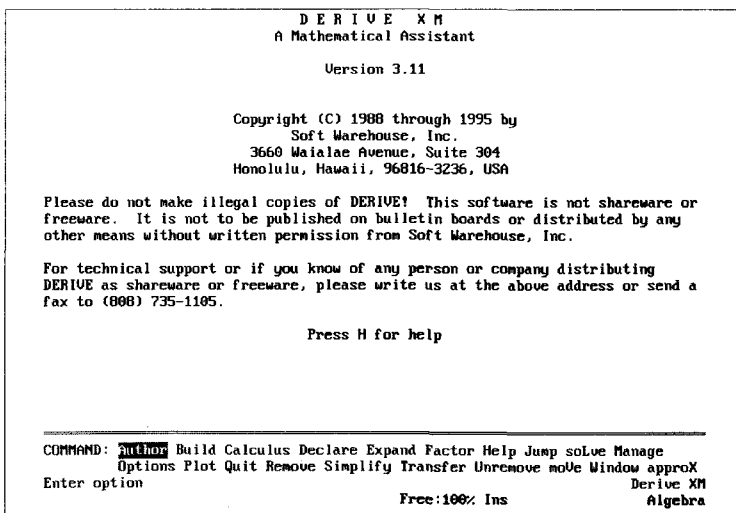
После загрузки Derive на экране дисплея появляется переднее панно системы (см. рис. 1.2). Оно содержит название системы, номер ее версии, данные о фирме-разработчике и предупреждение о нежелательности использования нелегальных копий Derive.

Далее пользователю напоминает, что нажатие клавиши H (от Help — справка) позволяет ввести режим справки. Это рекомендуется сделать уже при первом знакомстве с системой. Кстати, для русифицированной версии Derive раздел Справка выполнен на русском языке, что заметно облегчает работу нашим пользователям. Оригинальная версия Derive имеет, естественно, англоязычный справочный раздел. Поскольку команды главного меню одинаковы как для русифицированных, так и англоязычных версий, в дальнейшем мы не будем обращать внимание на тонкости русификации, если, конечно, они не имеют принципиального значения.

Двойная черта разделяет экран на два окна. Верхнее большое окно используется для ввода математических выражений и комментариев, а также для вывода результатов вычислений в символьном или числовом виде. Будем называть его окном документов.

Нижнее окно (окно меню) содержит всего четыре строки. Две из них — перечень позиций (команд) основного меню системы. В третьей строке — краткая подсказка. Вначале это слова “Enter option” (“Введите опцию”). Четвертая строка несет информацию о системе: номер строки выражения, имя загруженного файла, объем свободной памяти (в процентах от максимального), наличие или отсутствие режима вставки (Insert) и текущий режим работы системы.

Рис. 1.2
Переднее панно
системы
Derive 3.11 XM



Главное меню содержит следующие команды:

- Author** — ввод автором (пользователем) математических выражений для их последующего преобразования;
- Build** — построение математических выражений по их отдельным фрагментам;
- Calculus** — вычисление производных, интегралов, пределов функций, сумм, произведений и разложение в ряд Тейлора;
- Declare** — декларация (задание) функций, переменных, матриц и векторов;
- Expand** — раскрытие выражений, перемножение его членов, приведение к порядку по степеням;
- Factor** — факторизация чисел, разложение выражений на простые множители;
- Help** — включение справочной системы с кратким описанием возможностей Derive;
- Jump** — переход к заданному выражению;
- soLve** — решение математических уравнений;
- Manage** — управление вычислительным процессом;
- Options** — задание различных опций (установок) системы;
- Plot** — управление графическими средствами системы;
- Quit** — выход из системы (окончание работы с ней);
- Remove** — удаление ряда строк (или одной строки) из окна документов;
- Simplify** — упрощение выражений, выполнение подстановок, вычисление в рациональном виде;
- Transfer** — работа с файлами и принтером;
- move** — перенос блоков документов;
- Window** — работа с окнами;
- appoX** — приближенное вычисление выражений с применением численной аппроксимации результатов.

Одна из позиций основного меню (вначале Author) выделена цветовым фоном (или яркостью при монохромном дисплее). Чтобы сделать ее активной, нужно нажать клавишу ENTER. Дальнейшие действия зависят от того, какая позиция основного меню активна. Например, если активна позиция Quit, то произойдет (после подтверждения запроса о выходе) выход из системы в MS-DOS или в среду администратора диска Norton Commander. В других случаях (например, если активна позиция Calculus) появляется дополнительное подменю с рядом команд.

Пользователи современных ПК наверняка будут разочарованы тем, что версии Derive под MS-DOS вообще не поддерживают работу с привычной и удобной мышью — графическим манипулятором, передвигающим по экрану стрелку или иной указатель. Если использование мыши слишком привычно для вас, чтобы от него отказаться, переходите к работе с Derive 4.02 под Windows — эта система описана в главе 4 нашей книги. А пока настройтесь на управление только с помощью клавиатуры.

Выделение одной из позиций главного меню может производиться следующими клавишами:

Tab, BackSpace, Ctrl + I — перемещение вправо;

Ctrl + H, Ctrl + B — перемещение влево.

Здесь знак + означает совместное нажатие двух относящихся к нему клавиш.

Альтернативный и более быстрый способ выбора нужной позиции меню заключается в нажатии выделенной (большой) буквы имени позиции. В боль-

шинстве случаев (но не всегда) это первая буква имени. Тогда соответствующая команда выполняется сразу и не требуется нажатия клавиши ENTER. Этот способ является наиболее предпочтительным.

Поведение системы при выборе одной из команд основного меню может быть различным. Есть команды, которые выполняются сразу. Например, команда Autor включает в действие строчный редактор и позволяет сразу начать ввод вычисляемого выражения. Команда approx (аппроксимация) позволяет вычислить это выражение, а команда Simplify (упрощение) — упростить его.

Другие команды вызывают появление дополнительных подменю. Например, команда Transfer вызывает подменю, содержащее команды, относящиеся к работе с файлами: Merge (подключение файла к текущему документу), Clear (полная очистка системы), Load (загрузка файла с диска), Save (запись файла на диск), Print (вывод данных на печать), State (запись на диск информации о системе). В свою очередь, эти команды могут также вызывать подменю.

Работа с командами основного меню реализует большую часть возможностей системы (но не все). Ввиду простоты такой работы с ней и следует начинать освоение системы. Подробное описание команд основного меню и подменю дается в последующих разделах этой главы.

1.4. Редактирование и ввод выражений (Autor)

1.4.1. Ввод выражений

Прежде чем начинать какие-либо вычисления, необходимо задать (вести) одно или несколько математических выражений, над которыми предполагается выполнение вычислений или математических преобразований. Они могут быть простыми (например $2 + 3$) или сложными — в виде громоздкой математической формулы, содержащей различные математические функции, такие, как $\sin(x)$, $\exp(x)$ или функции Derive. В последнем случае принципиально необходимы средства редактирования выражений.

Математические выражения состояются из операторов, чисел, имен переменных и констант и функций. Операторами являются знаки математических операций, например $+$, $-$, $*$ и $/$; это арифметические операторы сложения, вычитания, умножения и деления. Они применяются совместно с операндами — числами или переменными. Например, в выражении $2 + 3$ числа 2 и 3 — это операнды, а знак $+$ — оператор.

Функции, например $\sin(x)$ или $\exp(x)$ и др., характерны тем, что они возвращают значение и имеют заданный в скобках параметр. Существуют функции нескольких параметров. Вот пример более сложного выражения: $(2 + 3) * \sin(x)$. Кстати говоря, функции в Derive могут вводиться буквами любого размера, так что $\sin(x)$ и $\sin(x)$ означает одно и то же — синус “икс”. Но Derive всегда записывает имя функции, например \sin , прописными буквами.

По умолчанию используются однобуквенные переменные, например x , y и z ; кроме того, знак умножения заменяется пробелом. Так что $x\ y\ z$ Derive преобразует в $x\ y\ z$ и будет вычислять произведение трех переменных x , y и z . Позже будет отмечено, как вводить переменные с многобуквенными именами, например $x\ y\ z$ или $X_coordinate$.

При записи математических выражений надо учитывать, что Derive понимает запись чисел во всех общепринятых форматах с применением для отделения мантииссы и порядка точки, а не запятой. При вводе можно использовать знак возведения в степень \wedge для записи чисел в научной нотации, например $123.456 \cdot 10^3$. Заметим, что это целое число 123456.

Кроме того, при записи математических выражений важное значение имеет приоритет операций. Он выше у сложных функций и ниже у простых математических операторов. В целом приоритет идет в следующем порядке: специальные функции, обычные математические функции, умножение и деление, сложение и вычитание. Для изменения приоритета используются круглые скобки. Например, при записи $(2 + 3)/5$ вначале будет выполнено сложение $(2 + 3)$, а затем уже деление. А в выражении $2 + 3/5$ вначале будет выполнено деление 3 на 5, а затем к результату будет прибавлено число 2.

Математические выражения вводятся с помощью команды Autor и соответствующей позиции главного меню. После ввода появляется запрос:

AUTHOR expression: <Авторское выражение>

Теперь нужно ввести какое-либо выражение, отредактировать его и зафиксировать ввод нажатием клавиши ENTER. Здесь и далее перевод сообщений на английском языке дается в скобках $\langle \rangle$ (он, естественно, на экран дисплея не выводится).

Введенное выражение окажется перенесенным в текущую нижнюю строку в верхней части экрана. Если система была только что загружена, то нумерация строк начинается с цифры 1. А если перед этим вы работали с системой, то номер текущей строки будет произвольным. Все примеры в этой книге даны при нумерации строк, начинающейся с единицы. Для обеспечения такой нумерации следует очистить систему с помощью команды Transfer основного меню и затем команды Clear.

Математические выражения, до фиксации их ввода нажатием клавиши ENTER, вводятся и редактируются в одной строке — строке ввода. Она расположена в нижней части экрана. Однако количество знаков в строке ввода не ограничено. Если выражение не помещается в строке, левая часть его уходит за пределы экрана и можно продолжать ввод. Вместо скрытой части выражения появляется знак треугольника (он напоминает, что часть выражения скрыта в специальном буфере редактора).

1.4.2. Клавиши редактирования и управления курсором

Редактор математических выражений в Derive самый примитивный и простой — строчный. Значит, любое, даже самое сложное выражение придется вводить в одной строке. Для этого необходимо знание ввода выражений в таком виде. Надо полагать, что пользователи математических систем уже знают, как осуществляется такой ввод, в частности, он характерен для всех языков программирования, например для Бейсика или Паскаля.

Главное, что надо знать при редактировании в строчном редакторе, это то, что вводимые символы располагаются в месте нахождения текстового маркера (курсора), имеющего вид вертикальной черты. Клавиша вставки Ins позволяет устанавливать два режима работы — вставки ново-

го символа или замещения им имеющегося символа справа от курсора. Клавиша стирания ← уничтожает символ слева от курсора, а клавиша Del — справа от курсора.

Клавиши перемещения курсора используются для перемещения по полю ранее введенных выражений текстового маркера. Для редактирования вводимых по команде Authorg математических выражений используются указанные ниже команды редактирования.

Специальные клавиши:

- Backspace** — удалить символ слева от курсора;
- Del** — удалить символ над курсором;
- Enter** — ввести строку текста;
- Ctrl + Enter** — ввести и упростить строку текста;
- Esc** — прервать редактирование и вернуться в меню;
- Ins** — переключить режим вставки/замены;
- F6** — переключить режим клавиш-стрелок (строчный редактор или выделение подвыражений);
- F3** — выделить выделенное выражение;
- F4** — вставить выделенное выражение, заключив его в скобки.

Команды управления курсором (в режиме выделения выражений клавиши-стрелки не работают):

- Ctrl + S** — влево на символ;
- Ctrl + D** — вправо на символ;
- Ctrl + A** — влево на слово;
- Ctrl + F** — вправо на слово;
- Ctrl + Q S** или **Home** — к левому краю строки;
- Ctrl + Q D** или **End** — к правому краю строки.

Команды удаления текста:

- Ctrl + H** — удалить символ слева от курсора (Backspace);
- Ctrl + G** — удалить символ над курсором (Del);
- Ctrl + T** — удалить слово; начинающееся над курсором;
- Ctrl + Y** — удалить всю строку;
- Ctrl + Q Y** — удалить правую часть строки;
- Ctrl + Q H** — удалить левую часть строки.

Прочие команды:

- Ctrl + M** — ввести строку текста (Enter);
- Ctrl + J** — ввести и упростить строку текста (Ctrl + Enter);
- Ctrl + [** — прервать редактирование и вернуться в меню (Esc);
- Ctrl + V** — переключить режим вставки/замены (Ins);
- Ctrl + U** — вставить предыдущую строку текста.

После нажатия клавиши ENTER выражение из командной строки переходит в окно редактора и занимает в нем соответствующую строку, точнее ряд строк с общим номером.

1.4.3. Выделение выражений и подвыражений

Любое выражение или часть его можно выделить. Выделенное выражение отображается инверсными по цвету знаками с инверсным фоном. Например, если обычно ваши знаки представляются белыми на черном фоне, то выде-

ленное выражение будет представлено черными знаками на белом фоне. Для управления областью выделения используются представленные ниже комбинации клавиш.

Команды выделения выражений:

Ctrl + E или стрелка вверх	– перейти вверх на одно выражение;
Ctrl + X или стрелка вниз	– перейти вниз на одно выражение;
Ctrl + R или PgUp	– перейти вверх на пол-экрана;
Ctrl + C или PgDn	– перейти вниз на пол-экрана;
Ctrl + PgUp or Ctrl + Home	– перейти вверх на первое выражение;
Ctrl + PgDn or Ctrl + End	– перейти вниз на последнее выражение;
Ctrl + F or Ctrl→	– передвинуть выражение влево на пол-экрана;
Ctrl + A or Ctrl←	– передвинуть выражение вправо на пол-экрана.

Команды выделения подвыражений:

→	– передвинуться на один операнд вправо;
←	– передвинуться на один операнд влево;
Home	– перейти к крайнему левому операнду;
End	– перейти к крайнему правому операнду;
Ctrl + E или стрелка вверх	– перейти на один уровень вверх;
Ctrl + X или стрелка вниз	– перейти на один уровень вниз.

Выделение выражений или подвыражений — важная часть работы с системой Derive. Поэтому настоятельно рекомендуем поупражняться в выделении частей какого-нибудь достаточно сложного выражения и освоить технику выделений его как целиком, так и по отдельным частям.

1.4.4. Примеры ввода и редактирования выражений

На рис. 1.3 представлен экран Derive после ввода двух математических выражений. Последнее из введенных выражений выделяется: оно размещается в четырехугольнике и представлено в инверсном цвете. Как уже отмечалось, область выделения можно менять.

Их ввод в командной строке после исполнения команды Autor выглядит следующим образом:

$$(a \cdot x^2 + b \cdot x + c) / (2 \cdot a \cdot b \cdot c \cdot x)$$

и

$$\sin(x) / (x \cdot (\sin(x)^2 + \cos(x)^2))$$

Ввод завершается нажатием клавиши Enter, после чего выражения в преобразованном виде появляются в окне выражений (или в алгебраическом окне). С выделенным выражением можно выполнять различные операции. Например, на рис. 1.4 показан результат выполнения операции Simplify (упростить) над выражением #2 (знак # используется для указания выражения).

Рис. 1.3
Ввод двух
математических
выражений

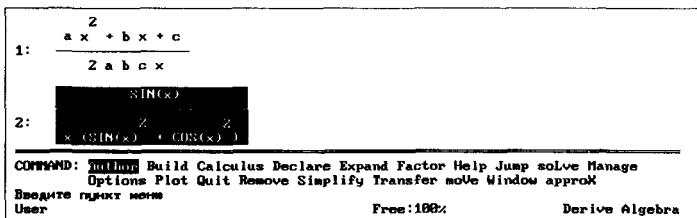


Рис. 1.4
Выполнение
операции
Simplify над
выражением #2

1:
$$\frac{2ax^2 + bxc}{x(\sin(x)^2 + \cos(x)^2)}$$

2:
$$\frac{\sin(x)}{x}$$

3:
$$\frac{\sin(x)}{x}$$

CALCULUS: **Differentiate** Integrate Limit Product Sum Taylor

Время счета: 0.0 секунд
User Free:100% Derive Algebra

Рис. 1.5
Задание
вычисления
производной
знаменателя #1

1:
$$\frac{2ax^2 + bxc}{x(\sin(x)^2 + \cos(x)^2)}$$

2:
$$\frac{\sin(x)}{x}$$

3:
$$\frac{\sin(x)}{x}$$

4:
$$\frac{1}{dx} (2ax + b)$$

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove **Simplify** Transfer move Window approx

Введите пункт меню
Dif(1',x) Free:100% Derive Algebra

Обычно выделена последняя строка в окне выражений. Однако с помощью указанных выше клавиш можно перемещать выделение с одной строки на другую и даже с одной части выражения на другую. Например, чтобы выделить знаменатель выражения #1, достаточно нажимать клавишу перемещения курсора вверх. Результат показан на рис. 1.4.

С выделенным выражением можно выполнять различные операции. На рис. 1.5 представлен случай, когда задается дифференцирование выделенной на рис. 1.3 части выражения, — используется команда Differentiate в позиции Calculus главного меню.

Если теперь исполнить команду Simplify, будет получена производная знаменателя выражения #1, представленная к вычислению выражением #4 (см. рис. 1.6).

Так выполняются типовые вычисления в системе Derive. В дальнейшем будет приведено множество других примеров проведения вычислений. Не плохо бы и пользователям Derive под MS-DOS просмотреть примеры, приведенные в главе 4, где описана куда более комфортная работа в версии Derive 4.02 под Windows.

1.4.5. Ввод греческих букв и проблема русификации Derive XM

Derive (без загрузки драйверов кириллицы) позволяет использовать в математических выражениях греческие буквы. Для этого совместно с клавишей Alt используются следующие клавиши:

Alt + A	alpha	альфа	(р)	Alt + M	mu	мю	(ц)
Alt + B	beta	бета	(с)	Alt + P	pi	пи	(у)
Alt + G	gamma	гамма	(т)	Alt + S	sigma	сигма	(х)
Alt + D	delta	дельта	(ы)	Alt + T	tau	тау	(ч)
Alt + N	epsilon	эпсилон	(ю)	Alt + F	phi	фи	(ш)
Alt + H	theta	тэта	(щ)	Alt + O	omega	омега	(ъ)

Полезно также знать клавиши для ввода некоторых распространенных констант и функции квадратного корня:

Alt + E #e — основание натурального логарифма (И);

Alt + I #i — квадратный корень из -1 (М);

Alt + Q SQRT — функция квадратного корня.

ASCII-коды букв греческого алфавита используются для задания символов кириллицы при обычной альтернативной кодировке знакогенераторов ви-

Рис. 1.6
Вычисление
производной по
выражению #5

1:
$$\frac{a x^2 + b x + c}{2 a b c x} \sin(x)$$

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

3:
$$\frac{\sin(x)}{x}$$

4:
$$\frac{d}{dx} (a x^2 + b x + c)$$

5:
$$2 a x + b$$

COMMAND: **Autor** Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer move Window approx
Время счета: 0.0 секунд Free:100% Derive Algebra
Simp(4)

деоадаптера дисплея и принтера. Поэтому после загрузки драйвера дисплея использовать буквы греческого алфавита уже нельзя: вместо них появятся соответствующие символы кириллицы (они указаны в таблице букв в круглых скобках). Поэтому знаки кириллицы следует применять только для записи комментариев на русском языке. Вообще говоря, если для вас главное — корректная запись математических выражений, следует отказаться от русификации Derive и загрузки драйвера русскоязычных шрифтов, перейдя к применению оригинальной англоязычной версии Derive.

Математические выражения в системе Derive задаются в общепринятой для большинства языков программирования алгебраической форме записи. Они содержат константы, числа, операторы (например, знаки арифметических операций, скобки и т. д.) и функции.

Операторы указывают на то, какие операции проводятся с их параметрами, называемыми операндами. Например, в выражении $2 + 3$ оператор $+$ указывает на сложение операндов — чисел 2 и 3. Функция $\sin(1)$ возвращает значение преобразованного параметра (в этом примере число 1 — фактический параметр функции, а $\sin(1)$ возвращает синус 1 рад.). Поэтому функции можно использовать в составе математических выражений, например $\exp(2 * \sin(1))$.

Скобки используются для задания приоритета выполнения операций. Они и арифметические операторы являются разделителями, поэтому можно не вводить пробелов при наборе выражений — при вводе в окно документов Derive автоматически расставляет пробелы, представляя выражение в удобном для просмотра виде.

1.4.6. Примеры работы с системой Derive XM

При работе с математическими пакетами всегда встречается противоречие между естественной (или, точнее, общепринятой) формой записи выражения и набором символов, которые может вводить клавиатура ПК и отображать его дисплей. Например, на клавиатуре нет знаков квадратного корня, производной, интеграла и пр., хотя знакогенератор дисплея частично эти знаки содержит. Поэтому пользователи ПК привыкли к различным обозначениям, например, $\exp(x)$ для числа e в степени x , $\text{sq}(x)$ или $\text{sqrt}(x)$ для вычисления квадратного корня x , x^n для x в степени n и т. д.

Это стало настолько привычным, что появление на экране дисплея естественных математических символов (как, например, при работе системы MathCAD [4,5]) воспринимается как особый шик, даже если ввод их требует использования необычных и подчас трудно запоминаемых комбинаций клавиш.

Derive решает эту проблему просто и изящно. Ввод математических выражений производится так, как в большинстве языков программирования. А вот отображение выражений на экране дисплея (как входных, так и результатов вычислений) максимально приближено к их общепринятому виду. Рассмотрим несколько примеров.

Пусть имеются две точки с координатами (x, y) соответственно равными $(2.3, 4)$ и $(8.5, 0.7)$. Введя команду Autor, запишем в строке ввода следующее выражение:

$$\text{sqrt}((2.3 - 8.5)^2 + (4 - 0.7)^2)$$

Нажав теперь клавишу Enter, увидим это выражение в строке под номером 1:

$$1: \sqrt{((2.3 - 8.5)^2 + (4 - 0.7)^2)}$$

Обратите внимание, что слово sqrt исчезло и вместо него появился знак квадратного корня $\sqrt{\quad}$. При этом для распространения его действия на все выражение используются общие круглые скобки.

Исчез и знак возведения в степень $^{\quad}$ — эта операция также представлена в общепринятом виде. Впрочем, повод для замечаний есть; например, знак квадратного корня не охватывает все выражение (ничего страшного тут нет, поскольку выражение стоит в круглых скобках). Далее мы увидим, что знак умножения $*$ в окне документов тоже исчезает и вместо него стоит пробел ($a*b$ будет отображено в виде $a b$).

Теперь чтобы вычислить это выражение, достаточно воспользоваться командой приближенного вычисления выражений arроX (для этого достаточно нажать клавишу X). В ответ появится запрос:

APPROX expression: #1 <Вычисляется выражение: #1>

Пока ограничимся очевидным и нажмем клавишу ENTER — будет вычислено выражение в строке 1 и результат вычислений появится в строке 2.

2: 7.02352

Теперь задайте такое арифметического выражение:

$$2+(3+4)/(5-1)$$

На экране дисплея появится:

$$3: 2 + \frac{3+4}{5-1}$$

4: 3.75

Обратите внимание, что знак деления в виде наклонной черты / оказался замененным на горизонтальную черту. Полезно обратить внимание, что выражение в строке #3 построено по общепринятым приоритетам математических операций с учетом действия изменяющих приоритет скобок.

Последний пример демонстрирует вычисление факториала:

5: 10!

6: 3.6288 10⁶

Результат оказался представленным в не совсем удобном виде числа с плавающей точкой (как и в большинстве языков программирования, дробная часть отделяется от целой точкой, а не запятой). В дальнейшем будет показано, что Derive позволяет задавать различные форматы представления чисел, в том числе рациональный и целочисленный, желательный для этого примера.

При использовании команды arроX для вывода результатов в форме чисел и при применении ряда других команд необходимо указывать, к какому именно выражению они применяются. Это не обязательно должно быть выражение, стоящее в последней строке документа. Для указания выражения используется символ #, после которого и указывается номер строки с нужным выражением.

Важное достоинство системы — возможность проведения вычислений в символьном виде. Ниже показан пример таких вычислений.

1: "Зададим выражение (команда Autor):"

$$2: \text{SIN}(x)^2 + \text{COS}(x)^2$$

3: "Вычисляем выражение #2 (команда arроX)"

4: 1

5: "Выполним разложение на простые дроби выражения"

$$6: \frac{x^2 - 5}{x(x-1)^4}$$

7: "Разложим #6 на простые дроби (команда Expand):"

$$8: \frac{4}{(x-1)^4} + \frac{6}{(x-1)^3} - \frac{5}{(x-1)^2} + \frac{5}{x-1} - \frac{5}{x}$$

9: "Для выражения #8 используем команду Simplify:"

$$10: \frac{5x^3 - 20x^2 + 31x - 20}{(x-1)^4} - \frac{5}{x}$$

11: "Для #8 используем команду Factor Racional:"

$$12: \frac{x^2 - 5}{x(x-1)^4}$$

13: "Для #10 используем команду Factor Racional:"

$$14: \frac{x^2 - 5}{x(x-1)^4}$$

В строке 2 представлено тригонометрическое выражение, вводимое в виде:

$$\sin(x)^2 + \cos(x)^2$$

Поскольку значение x не определено, то в большинстве математических систем для численных расчетов попытка вычислить это выражение привела бы к появлению сообщения об ошибке. Однако система Derive способна проводить вычисления и в этом случае. Для нее появление неопределенной переменной — сигнал к переходу к символьным вычислениям. Как и школьник, она "знает", что при любом x сумма квадратов синуса и косинуса равна 1. Что и подтверждается в строке 4, полученной при использовании команды arроX или Simplify.

Далее в строке 6 введено, казалось бы, простое алгебраическое выражение. Для его ввода с клавиатуры надо набрать:

$$(X^2 - 5) / (x * (x - 1)^4)$$

Для разложения этого выражения на простые дроби достаточно исполнить команду Expand. Для не умудренного знанием математики человека результат, увиденный в строке 8, может показаться ошеломляющим: оказывается, простое выражение #6 раскладывается на целых пять членов! И таких примеров можно привести великое множество.

Возникает вопрос: а можно ли сложное выражение, например #6, свести к более простому виду? Ответ положительный. В строках 9 и 10 показано применение команды Simplify к выражению #8. Нетрудно заметить, что выражение #8 упростилось, но вовсе не приняло первоначальный вид. Это связано со спецификой команды Simplify. Чтобы предельно упростить выражение #8, надо использовать для него команду рациональной факторизации Factor Racional, что и сделано в строках 11 и 12. В итоге сложное выражение #8 приобрело простой исходный вид. В строках 13 и 14 показано применение этой команды для промежуточного выражения #10. И здесь оно преобразовано к исходному виду #8.

Итак, мы рассмотрели три формы одного и того же алгебраического выражения. В действительности этих форм может быть намного больше. Не сле-

дует полагать, что любое выражение из математического справочника Derive тут же приведет к аналогичному тому, который дан в справочниках [10, 11]. Так будет только в том случае, когда результат преобразований однозначен и доступен Derive, например в случае вычисления выражения в строке 2. При неоднозначном результате Derive, как и любой математик, имеет право на “самостоятельное” творчество и может выдать результат преобразования порою в довольно неожиданном виде. Если же преобразование выражения вообще не по зубам системе, выражение будет просто повторено в последней строке.

Такая ситуация обычно возникает из-за отсутствия во встроенной в ядро Derive библиотеке необходимых функций и правил преобразования. Библиотека Derive создана весьма удачно. Набор ее функций (их порядка 1000) удовлетворяет большинство пользователей. Но он не столь обширен, как в библиотеках таких монстров компьютерной алгебры, как Maple V или Mathematica 2/3, и ориентирован главным образом на преобразования выражений с элементарными функциями. Именно поэтому пользователи — математики высшей квалификации отдают предпочтение этим громоздким пакетам, несмотря на их чрезмерные запросы по ресурсам ПК, медлительность в работе и даже возможные ошибки.

Текст, представленный в последних примерах (и в большинстве последующих), нельзя назвать программой в общепринятом смысле этого слова. Скорее это одновременная запись текстовых комментариев, математических выражений и алгоритмов решения задач, а также результатов преобразования выражений и решения задач как в символьном, так и численном виде. Поэтому, как и в системе MathCAD, этот текст в нашей книге называется документом, а не программой. Как уже говорилось, он отражает рабочую сессию.

При решении многих задач система Derive не требует от пользователя знания даже основ программирования. Достаточно описать алгоритм решения задачи в виде совокупности математических выражений, т. е. уметь составлять документ для решения нужных задач. Поэтому систему Derive относят к системам без программирования. В главе 2 будет показано, что это не совсем правильно и при необходимости пользователь может запрограммировать решение своих специфических задач на входном языке системы сверхвысокого уровня.

В дальнейшем, за исключением главы 4, мы будем приводить большинство примеров в виде распечатки полного текста текущего документа. Он создается командами Transfer Print File в виде файла в текстовом формате с расширением .prt. Программный файл с расширением .mth также записывается в текстовом формате. Однако записи в нем хранятся в том виде, в котором осуществляется ввод по команде Autor. Более подробное описание этих файлов дано в разделе 1.19 настоящей главы.

1.5. Построение математических выражений (Build)

При записи математических выражений часто оказывается, что они содержат фрагменты ранее введенных выражений или повторяют полностью такие выражения. При этом для включения их в редактируемое выражение могут потребоваться дополнительные соединительные знаки и операторы или специальные указания (например, об инвертировании используемого фрагмента). Эти возможности реализуются с помощью команды Build (построение).

При выполнении команды Build на месте главного меню появляется запрос:

BUILD first expression: #n <Первое выражение>

Здесь число n указывает, в какой строке находится это выражение. При этом для построения используется только выделенная часть выражения. Указав нужный номер n (или согласившись с предложенным системой), нужно нажать клавишу ввода ENTER.

Появится новый запрос:

**BUILD: Operator: + - * / ^ . ` = Minus Recip Ln
Exp Tan Sin Cos Atan ! % Done**

Этот запрос позволяет выбрать необходимый дополнительный оператор или задать необходимую функцию. Смысл обозначений операторов и функций очевиден, за исключением следующих:

- ` — транспонирование (для матриц);
- Minus** — смена знака у выражения;
- Recip** — задание обратной величины;
- !** — вычисление факториала;
- Done** — конец работы с командой Build.

Для выбора нужного оператора или функции надо перемещать световую метку, выделяющую одну из позиций, с помощью клавиши пробела BackSpace (вправо) и Ctrl + H (влево). Завершается выбор нажатием клавиши ENTER. При этом появится еще один запрос:

BUILD next expression: #n <Следующее выражение>

Теперь необходимо выделить следующее выражение. Номер n меняется автоматически, когда клавишами перемещения курсора отмечается та или иная строка. Выбрав нужную строку и обозначив в ней выделенный фрагмент, нужно нажать клавишу Enter. Такие действия повторяются, пока не будет задано выражение. По окончании конструирования выражения необходимо в перечне операторов и функций указать команду Done. В ответ выражение окажется в соответствующей строке, и в нижнем окне вновь появится главное меню. Полезно напомнить, что нажатие клавиши F3 переносит выделенное выражение в строку ввода без скобок, а F4 — в скобках.

Рассмотрим примеры на конструирование выражений. Очистим верхнее окно с помощью команды Clear (нажав клавиши T и C). Теперь введем две строки с выражениями $\sin(2 \cdot x)$ и $\cos(x)$. На экране дисплея появится:

```
1: SIN (2 X)
2: COS (X)
```

По первому запросу Build выделим в строке 1 выражение 2 X. По второму запросу укажем оператор *. По третьему запросу укажем целиком выражение в строке 2. По четвертому запросу укажем команду Done. В итоге получим в строке 3 следующее выражение:

```
3: 2 X COS(X)
```

Несколько отвлечемся от описания команд только главного меню и покажем, как можно вычислить численное значение выражения в строке 3 при $X = 1.5$. Для этого воспользуемся позицией Manage (нажав клавишу M) и да-

лее командой Substitute (нажав клавишу S). При этом надо указать номер строки вычисляемого выражения (в нашем случае n). Появится запрос:

MANAGE SUBSTITUTE value: <Значение>

В ответ на запрос надо указать значение 1.5. В строке 4 появится выражение:

4: 2 1.5 COS (1.5)

Нетрудно заметить, что в результате этих действий система подставила в исходное выражение вместо переменной X численное значение 1.5. Теперь, используя команду appoX (нажатием клавиши X) и указав номер строки 4 с вычисляемым выражением, можно, нажав клавишу Enter, получить результат вычислений:

5: 0.212212

Он помещается в строку 5. Таким образом можно подставить в формулу и заданное символьное выражение.

1.6. Специальные вычисления (Calculus)

Позиция Calculus главного меню содержит ряд операторов, реализующих специальные вычисления, присущие достаточно сложным математическим системам. Они реализуют наиболее распространенные операции высшей математики. При активизации этой команды появляется подменю, в котором перечислены функции этих операторов:

Differentiate — вычисление производных;
Integrate — вычисление интегралов;
Limit — вычисление пределов функций;
Product — вычисление произведения членов ряда;
Sum — вычисление суммы членов ряда;
Taylor — разложение функции в ряд Тейлора.

Большинство этих вычислений достаточно трудоемко и требует хорошего знания математики. Derive облегчает их проведение, но не исключает необходимости в понимании их математической сути. При этом надо особо отметить, что Derive выполняет такие вычисления не только в численном виде, но и в символьном. Более того, именно последнее является отличительной чертой системы Derive, относящейся к системам компьютерной алгебры.

1.6.1. Вычисление производных

Порядок вычисления производных рассмотрим на ряде примеров. Пусть требуется найти третью производную функции $\sin(2x)$. Вначале, используя команду Author, введем это выражение (считаем, что предварительно была введена команда очистки Clear).

Далее выбираем команду дифференцирования выражений Differentiate. По запросу

CALCULUS DIFFERENTIATE expression: #n <Выражение>

укажем (если нужно) номер строки n, равный 1. По следующему запросу

CALCULUS DIFFERENTIATE variable: x <Переменная>

необходимо указать, по какой переменной будет производиться дифференцирование (в общем случае функция может быть функцией ряда переменных).

В данном случае нас устроит указание переменной x по умолчанию.

Наконец, по последнему запросу

CALCULUS DIFFERENTIATE Order: 1 <Порядок>

нужно указать порядок производной: 3 вместо 1 по умолчанию.

Теперь во второй строке появится выражение с описанием производной.

Далее, воспользовавшись командой arproX, найдем значение третьей производной в символьном виде: $-8 \cos(2x)$. Документ, полученный в результате этих операций, представлен ниже:

1: SIN (2 x)

2: $\frac{d^3}{dx^3}$ SIN (2 x)

3: - 8 COS (2 x)

Рассмотрим еще ряд практических примеров численного дифференцирования:

1: "Символьное дифференцирование - используется команда Simplify"

2: "Вычисление первой производной функции a^x "

3: $\frac{d}{dx} a^x$

4: $a^x \ln(a)$

5: "Вычисление первой производной функции asinh(x)"

6: $\frac{d}{dx} \text{ASINH}(x)$

7: $\frac{1}{\sqrt{x^2 + 1}}$

8: "Вычисление третьей производной функции x^a "

9: $\frac{d^3}{dx^3} a^x$

10: $a^x \cdot a^{-3} (a - 2)(a - 1)$

11: "Вычисление частных производных"

12: $F(x, y) := x^3 y^4$

13: $\left(\frac{d}{dy} \right)^3 \left(\frac{d}{dx} \right)^2 F(x, y)$

14: 144 x y

Разумеется, Derive позволяет при необходимости найти производную и в численном виде для заданного значения аргумента дифференцируемой функции. Это иллюстрирует следующий пример:

1: "Вычисление производных в численном виде"

2: $\frac{\text{SIN}(x)}{x}$

3: $\frac{d}{dx} \frac{\text{SIN}(x)}{x}$

4: $\frac{\text{COS}(x)}{x} - \frac{\text{SIN}(x)}{x^2}$

5: "Для вычисления численного значения выражения #4"

6: "используйте команду Substitute в позиции Manage"

7: $\frac{\text{COS}(0.5)}{0.5} - \frac{\text{SIN}(0.5)}{0.5^2}$

8: "Затем используйте команду approx"

9: $-\frac{1207}{7426}$

В этих примерах вычисляется численное значение производной. Для этого используется команда подстановки Substitute в позиции Manager главного меню. После замены x на численное значение ($x = 0.5$) используется команда вычисления approx.

1.6.2. Интегрирование выражений

При вычислении в символьном виде интегралов (т. е. нахождении их первообразных) используется команда Integrate. Перед выдачей результата система задает запросы:

CALCULUS INTEGRATE expression: #n <Номер выражения>

CALCULUS INTEGRATE variable: x <Переменная>

CALCULUS INTEGRATE Lower limit: Upper limit <Пределы>

Если пределы (нижний — Lower и верхний — Upper) не указывать (нажать в ответ на запрос клавишу Enter), то будет вычислена первообразная для неопределенного интеграла в символьном виде. Если пределы указаны, то вычисляется значение определенного интеграла. Ниже приводится множество задач на взятие и вычисление интегралов в символьной форме.

1: "Вычисление интегралов в символьной форме"

2: "Используется команда Simplify"

3: "Вычисление интеграла функции $\sin(x)^4/\cos(x)$ "

$$4: \int \frac{\sin^4(x)}{\cos(x)} dx$$

$$5: \ln \left| \frac{\sin(x) + 1}{\cos(x)} \right| - \frac{\sin^3(x)}{3} - \sin(x)$$

6: "Вычисление интеграла функции $x \cdot \ln(a+bx)$ "

$$7: \int x \ln(a + bx) dx$$

$$8: \left(\frac{x^2}{2} - \frac{a^2}{2b} \right) \ln(bx + a) - \frac{x(bx - 2a)}{4b}$$

9: "Вычисление двойного интеграла"

10: $F(x, y) := x y$

$$11: \int_0^r \int_0^{\sqrt{r^2 - x^2}} F(x, y) dy dx$$

$$12: \frac{r^4}{8}$$

13: "Вычисление свертки"

14: "Пусть заданы функции:"

15: $F(x) := a x$

$$16: G(x) := m \left(1 - \exp \left(- \frac{x}{\tau} \right) \right)$$

17: "Тогда интеграл свертки есть:"

$$18: \int_0^t F(t-x) G(x) dx$$

$$19: \frac{a m (t^2 - 2 t \tau + 2 \tau^2)}{2} - a m \tau^2 e^{-t/\tau}$$

20: "Вычисление несобственного интеграла (b=inf)"

$$21: \int_0^{\text{inf}} x \exp(-x) dx$$

22: 1

23: "Вычисление несобственного интеграла (f(a)=inf)"

$$24: \int_0^1 \frac{1}{\text{SQRT}(x)} dx$$

25: 2

26: "Неберущиеся интегралы"

$$27: \int \frac{\text{SIN}(x)}{x} dx$$

$$28: \int \frac{\text{SIN}(x)}{x} dx$$

$$29: \int \exp(x) \text{LOG}(x) dx$$

$$30: e^x \text{LN}(x) - \int \frac{e^x}{x} dx$$

В строках 13—19 задано вычисление так называемого интеграла свертки, широко применяемого в электротехнических расчетах. А в строках 20—25 даны примеры вычисления несобственных определенных интегралов, т. е. таких, у которых либо предел стремится к бесконечности (inf), либо к бесконечности устремляется подынтегральная функция $f(x)$ в пределах интегрирования.

Вообще следует учитывать, что Derive вычисляет только те интегралы, которые выражаются в элементарных функциях. Могут встречаться неберущиеся интегралы, которые не выражаются в элементарных функциях. Например, интеграл, вычисляемый в строках 27 и 28. Здесь Derive отказалась взять интеграл и просто повторила его. Однако в случае, показанном в строках 29 и 30, интеграл представлен в виде, содержащем другой (тоже неберущийся) интеграл. Могут возникнуть трудности и со взятием интегралов от произвольных дробно-рациональных функций. Тогда их полезно разложить на простые дроби.

Приведенные в главах 1—3 этой книги распечатки примеров получены с использованием команд Transfer, Print и File. Они обеспечивают вывод полного содержания текущего документа в файл, имеющий текстовый формат. Этот файл (с расширением .prt) можно затем распечатать на любом принтере. Интерпретация записи примеров для Derive 3.11 была описана выше.

Ниже дан пример вычисления численных значений определенных интегралов.

1: "Вычисление определенного интеграла"

2: SQRT (2 x + 1)

3:
$$\int_0^1 \text{SQRT}(2x + 1) dx$$

4:
$$\frac{59984}{42885}$$

5: 1.39871

Если вычисление приводит к результату в дробно-рациональной форме, то для получения результата в форме обычного десятичного числа достаточно использовать команду arроX (что и сделано в конце приведенного примера).

1.6.3. Вычисление пределов функций

Для вычисления пределов функции в заданной точке используется оператор Limit. При этом надо ответить на следующие запросы:

CALCULUS LIMIT expression: #n <Номер выражения>

CALCULUS LIMIT variable: x <Переменная>

CALCULUS LIMIT Point: From: Both Left Right <Точка>

Первые два запроса определяют номер строки с выражением, задающим функцию, и переменную, по которой ищется предел. Последний запрос определяет, в какой точке (Point) ищется предел. После слова Point надо указать численное значение переменной, при котором нужно найти предел. Наконец, после слова From следует указать, как искать предел: с двух сторон (Both), слева (Left) или справа (Right) от выбранной точки. Ниже представлен пример вычисления пределов функций.

1: "Вычисление пределов функций"

$$2: \frac{\text{LN}(1+x)}{x}$$

$$3: \lim_{x \rightarrow 0} \frac{\text{LN}(1+x)}{x}$$

$$4: 1$$

$$5: \frac{1 - \text{COS}(x)}{x^2}$$

$$6: \lim_{x \rightarrow 0} \frac{1 - \text{COS}(x)}{x^2}$$

$$7: 0.5$$

$$8: \frac{2x - 2}{x - 1}$$

$$9: \lim_{x \rightarrow 1} \frac{2x - 2}{x - 1}$$

$$10: 2$$

Если предел ищется слева или справа от заданной точки, то это указывается знаками $-$ и $+$ после координаты точки. Предел при x , стремящемся к бесконечности, можно задать значением inf .

1.6.4. Вычисление произведения членов ряда

При вычислении произведений членов некоторого ряда надо вначале задать математическое выражение, описывающее произвольный член ряда. Затем, используя команду Product, можно найти произведение, ответив на следующие запросы:

CALCULUS PRODUCT expression: #n <Номер выражения>

CALCULUS PRODUCT variable: x <Переменная>

CALCULUS PRODUCT Lower limit:1 Upper limit:n <Пределы>

Если отказаться от ввода пределов (нажать клавишу ENTER), то произведение членов ряда будет вычислено в символьном виде. Следующий пример иллюстрирует это:

1: "Вычисление произведений членов ряда"

2: n

3: $\prod_{n=1}^n n$

4: n!

5: $\prod_{n=1}^{10} n$

$$6: \quad 3.6288 \cdot 10^6$$

В случае, показанном в строках 3 и 4, Derive представила произведение чисел 1, 2, ..., n в символьном виде, т. е. в виде факториала. А в другом случае (строки 5 и 6) вычислено конечное значение произведения из 10 натуральных чисел.

1.6.5. Вычисление суммы членов ряда

Для вычисления суммы членов ряда используется оператор Sum. После задания выражения, определяющего произвольный член ряда, необходимо ответить на запросы системы:

CALCULUS SUM expression: #n <Номер выражения>
CALCULUS SUM variable: x <Переменная>
CALCULUS SUM Lower limit:1 Upper limit:n <Пределы>

Если отказаться от ввода пределов (нажать клавишу ENTER), то сумма членов ряда будет вычислена в символьном виде. Ниже показан пример вычислений сумм членов рядов.

1: "Вычисление сумм рядов"

$$2: \quad a q^n$$

$$3: \quad \sum_{n=1}^n a q^n$$

$$4: \quad \frac{a q^{n+1}}{q-1} - \frac{a q}{q-1}$$

$$5: \quad 2^n - 1$$

$$6: \quad \sum_{n=1}^n (2^n - 1)$$

$$7: \quad n^2$$

$$8: \quad x^2$$

$$9: \quad \sum_{x=1}^n x^2$$

$$10: \quad \frac{1}{6} n(n+1)(2n+1)$$

$$11: \quad \sum_{x=1}^5 x^2$$

$$12: \quad 55$$

В этом примере показано, что суммы (как и произведения) могут вычисляться в символьном виде, если именно так заданы пределы изменения аргумента. Если же пределы указаны в виде чисел, то вычисляется конкретное значение суммы (см. строки 11 и 12).

В файлах .prt с текстами документов обозначения произведения Π и суммы Φ заменены словами PROD и SUM, поскольку многие типы принтеров не выводят указанных знаков. Однако в нашем случае мы привели распечатки с этими символами (а точнее, с очень похожими на них русскими буквами) как пример изображения, видимого на экране дисплея.

1.6.6. Разложение функции в ряд Тейлора

Derive позволяет найти разложение функции в окрестностях заданной точки в ряд Тейлора с заданным количеством членов разложения. Для этого достаточно определить функцию и ответить на следующие запросы системы:

```
CALCULUS TAYLOR expression: #n    <Номер выражения>
CALCULUS TAYLOR variable: x      <Переменная>
CALCULUS TAYLOR Degree:5 Point:0 <Степень и точка>
```

В последнем запросе (два первых очевидны) параметр Degree указывает на степень полинома, а параметр Point на значение переменной, при котором ищется разложение в ряд Тейлора. Разложение функций в ряд Тейлора показано в следующем примере:

1: "Разложение функций в ряд Тейлора"

2: COS (x)

3: TAYLOR (COS (x), x, 0, 5)

$$4: \frac{x^4}{24} - \frac{x^2}{2} + 1$$

5: LN (1 + x)

6: TAYLOR (LN (1 + x), x, 0, 5)

$$7: 0.2 x^5 - 0.25 x^4 + \frac{1}{3} x^3 - 0.5 x^2 + x$$

$$8: \frac{x^5}{5} - \frac{x^4}{4} + \frac{x^3}{3} - \frac{x^2}{2} + x$$

Здесь даны различные формы представления ряда Тейлора. Они зависят от того, какая из команд (Simplify или arproX) используется для вычислений. Команда Simplify обычно задает вывод результатов вычислений в дробно-рациональном виде, а arproX — в виде обычных десятичных дробей.

Все функции, входящие в позицию Calculus, могут задаваться и с помощью ключевых слов — имен этих функций. Это описывается несколько позже (в главе 2). Рекомендуется просмотреть множество других примеров вычислений, описанных в главе 4.

1.7. Декларация новых определений (Declare)

Derive позволяет задавать (декларировать) следующие новые определения: функции пользователя, переменные, матрицы и векторы. Для этого используется позиция главного меню Declare, содержащая подменю Function Variable Matrix vector.

Ниже дано описание деклараций с соответствующими примерами.

1.7.1. Декларация функций

Позиция Function используется для задания функций пользователя, которые являются расширениями базового входного языка системы. Такие функции после их задания можно использовать в математических выражениях наряду со встроенными функциями. Напоминаем, что функции в ответ на обращение к ним с указанием назначенных параметров возвращают свое значение. Оно может быть численным или символьным.

Если активизировать позицию Function появится запрос:

DECLARE FUNCTION name:

В ответ надо ввести имя функции. Тогда появится запрос:

DECLARE FUNCTION value:

В ответ следует ввести математическое выражение, описывающее функцию. Рассмотрим пример на задание функции трех переменных.

Пусть надо задать функцию

$$f(x, y, z) = x + y * z$$

В ответ на первый запрос укажите f — это и есть имя функции. В ответ на второй запрос укажите $x + y * z$. В документе появится строка с записью:

$$F(x, y, z) := x + yz$$

Это и есть определение новой функции. Заметим, что его можно задать и без использования команды декларации новой функции, используя знак присваивания :=.

Обратите внимание, что имя функции представляется всегда прописными буквами (хотя задаваться может строчными). Список аргументов (в скобках после имени функции) формируется автоматически. В качестве знака присваивания используются символы :=, поскольку знак равенства = служит для иных целей (задания условий).

1.7.2. Декларация переменных

Позиция подменю Variable позволяет задавать новые переменные. Переменные — это области памяти, снабженные именами. В них хранятся значения переменных, численные или символьные.

При вводе команды Variable появляется запрос:

DECLARE VARIABLE name:

В ответ надо ввести имя переменной. Затем появится запрос:

DECLARE VARIABLE: Domain Value

При задании обычных числовых переменных выбирается опция Value, после чего по запросу

DECLARE VARIABLE Value:

следует ввести числовое значение переменной. Опция Domain служит для задания области определения переменной. Если эта опция введена, то появляется запрос

DECLARE VARIABLE: Positive Nonnegative Real Complex Interval

Надо указать, какой тип переменной задается: с положительными значениями (Positive), неотрицательными (Nonnegative), реальными (Real), комплексными (Complex) и с интервальными (Interval). Более подробно об областях определения переменных рассказано в главе 4.

После декларации переменной она становится определенной и ее можно использовать в выражениях. Имя переменной (и любого другого определения) может состоять из цепочки знаков, но начинаться с буквы. Можно включать в эту цепочку знак подчеркивания “_”; например, переменная может иметь имя number_iterations. Имена можно набирать как строчными, так и прописными буквами. Однако Derive представляет имена переменных, векторов и матриц строчными буквами, а имена встроенных и определяемых пользователем функций — прописными.

В отличие от первых версий Derive в версии 3.11 не предусмотрена декларация констант. В качестве констант могут использоваться переменные с заданными значениями. Однако надо помнить о различии между переменными и константами: значение константы не меняется, а значение переменной всегда можно изменить. Вы можете использовать переменные в качестве констант, но надо следить за тем, чтобы им не были присвоены новые значения.

Задать новую переменную можно и без использования позиции Declare главного меню. Достаточно ввести (в позиции AutoG) следующее выражение:

Имя_переменной := Значение_переменной

Здесь знак := означает операцию присваивания. Переменной с указанным слева именем присваивается значение, равное значению числа или арифметического выражения справа от этого знака. Значение переменной может быть и символьным. Более того, оно может быть и функцией!

1.7.3. Декларация матриц и векторов

Позиция Matrix подменю позволяет задать двумерный числовой массив — матрицу. Как известно [29], матрица состоит из ряда строк и столбцов. Если в матрице m строк и n столбцов, то говорят, что матрица имеет размерность m*n. В Derive матрицы могут содержать как численные, так и символьные элементы матриц. Например, матрица с размерностью 2*3

```
1 2 3
4 5 6
```

является матрицей с численными элементами, тогда как матрица

$$\begin{array}{ccc} a & a+b & a-b \\ a*b & a/b & b \end{array}$$

представляет собой матрицу с символьными элементами.

Каждый элемент матрицы занимает определенное положение, которое задается указанием номера его строки и номера столбца. Если отсчет номеров вести с единицы, то элемент последней матрицы $a*b$ расположится на второй строке в первом столбце. Если матрица имеет имя M , то этот элемент обозначается как $M2,1$.

Для задания матриц используется команда `Matrix`. При ее активизации появляется запрос:

DECLARE MATRIX: Row:3 Columns:3

В ответ нужно ввести число строк (`Row`) и столбцов (`Columns`) задаваемой матрицы или согласиться с их значениями 3, предлагаемыми по умолчанию. Затем по запросам

MATRIX element:

нужно последовательно ввести значения всех элементов матрицы.

Позиция `vector` подменю служит для задания вектора — матрицы, вырожденной в одну строку. При активизации позиции `vector` появляется запрос:

DECLARE VECTOR: Dimension

В ответ нужно указать размерность (`Dimension`) вектора, т. е. число элементов в векторе. Затем по запросам

VECTOR element:

следует ввести значения всех элементов вектора.

Векторы и матрицы можно вводить в текст документов и без использования позиции `Declare` главного меню. Для ввода вектора достаточно перечислить значения его элементов, разделяя их запятыми и заключая в квадратные скобки. Например:

`[1, 2, 3]` задает вектор с элементами 1, 2 и 3.

Аналогично (с использованием двойных квадратных скобок) можно задать матрицу. Например:

`[[1, 2, 3],[4, 5, 6]]` задает матрицу $\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{vmatrix}$

Следует отметить, что наряду с вектором-строкой встречаются векторы-столбцы. Их можно задавать следующим образом:

`[[1], [2], [3]]` задает вектор-столбец $\begin{vmatrix} 1 \\ 2 \\ 3 \end{vmatrix}$

Такие векторы встречаются, например, при решении систем линейных уравнений. Важно отметить, что результаты многих векторных и матрично-векторных операций различны при использовании в них векторов-строк или векторов-столбцов. Можно сказать, что векторы и матрицы задаются с помощью квадратных скобок как списки. Еще раз отметим, что элементы матриц и

векторов могут иметь как численное, так и символьное значение. Именно символьные операции с матрицами характеризуют мощь Derive как системы символьной математики.

1.8. Раскрытие выражений (Expand)

1.8.1. Техника применения команды Expand

Команда Expand служит для раскрытия (расширения) математических выражений, в основном полиномов. Эта команда стремится сделать максимальным число членов выражения, алгебраически независимых от переменных, по которым происходит расширение. Оно может идти по одной переменной или по ряду переменных. При этом по умолчанию задан определенный порядок переменных — x , y и z . Команда может применяться как к полному выделенному выражению, так и к его выделенным фрагментам. Это зачастую позволяет получать различные результаты преобразований.

В явном виде такая команда (и соответствующая ей функция) к тригонометрическим и алгебраическим выражениям не применяется — впрочем, если они не имеют скрытой полиномиальной формы, которая при раскрытии может измениться. Команда Expand не имеет вариантов. Если математическая формула не может быть раскрыта, т. е. уже представлена в раскрытом виде, то при использовании команды Expand она будет просто повторена в окне документов.

После пуска этой команды появляется запрос:

EXPAND expression: #n

Надо указать номер строки выражения n , подвергаемого раскрытию. Затем появляется запрос о том, по какой переменной происходит упорядочение выражения по степеням:

EXPAND variable n:

Если задать раскрытие по всем переменным, то в ответ на этот запрос следует просто нажать клавишу ENTER. Естественно, можно задать раскрытие по заданной переменной, указав ее по запросу.

Рассмотрим пример на преобразование выражений в символьном виде.

1: "Действие команды Expand"

2: "Исходное алгебраическое выражение"

3: $(a - b)(a + b)$

4: "Результат его преобразования"

5: $a^2 - b^2$

6: "Преобразование тригонометрических выражений"

7: $\frac{\sin(x)}{\cos(x)}$

- 8: TAN (x)
- 9: $\text{SIN}(x)^2 + \text{COS}(x)^2$
- 10: 1
- 11: $\text{SIN}(a) \text{COS}(b) - \text{COS}(a) \text{SIN}(b)$
- 12: "Этот результат вы получите только используя"
- 13: "опции Collect и Trigonom позиции Manager"
- 14: $\text{SIN}(a - b)$

Полезно обратить внимание на то, что результат преобразования зависит от заданных пользователем (позиция Manage) установок системы.

1.8.2. Синтез полинома по его действительным и комплексным корням

Интересными примерами применения команды Expand является синтез полиномов (степенных многочленов) по их действительным или комплексным корням.

- 1: "Синтез многочлена по его корням a, b и c"
- 2: $(x - a)(x - b)(x - c)$
- 3: "Полученный многочлен"
- 4: $x^3 - x^2(a + b + c) + x(a(b + c) + b c) - a b c$
- 5: "Синтез многочлена по его численным корням (1, 2, 3)"
- 6: $(x - 1)(x - 2)(x - 3)$
- 7: "Полученный многочлен"
- 8: $x^3 - 6x^2 + 11x - 6$
- 9: "Синтез полинома с комплексными корнями"
- 10: $(x + 1 + \#i 2)(x + 1 - \#i 2)(x + 3)(x + 4)$
- 11: $x^4 + 9x^3 + 31x^2 + 59x + 60$

Из приведенных примеров видно, что команда Expand — мощное средство проведения символьных преобразований. Особенно удобна она при работе с многочленами и дробно-рациональными функциями.

1.9. Факторизация (Factor)

1.9.1. Разложение чисел на простые множители

Как известно, многие целые числа могут разлагаться на простые множители. Этот процесс называется факторизацией и реализуется в системе Derive командой Factor. При ее вводе появляется запрос:

FACTOR expression: #n

Например, для разложения на простые множители числа 123456789 достаточно задать это число в какой-либо строке и ответить на указанный запрос. Разложение будет представлено в виде:

$3^2 3607 3803$.

Здесь число 3 во второй степени означает, что множитель 3 встречается дважды, т. е. $123456789 = 3 \cdot 3 \cdot 3607 \cdot 3803$.

1.9.2. Разложение на части выражений

Derive использует команду Factor и в более общем виде, с целью разложения произвольного математического выражения на его отдельные части, например простые множители. Если эта команда применяется не к числу, а к выражению, то появляется дополнительный запрос:

FACTOR: Amount: Trivial Squarefree Rational Radical Complex

Необходимо указать область (Amount) факторизации. Она может быть одной из следующих типов: обычная (Trivial), свободная от квадратов (Squarefree), рациональная (Racional), радикальная (Radical) и комплексная (Complex).

Область факторизации существенно влияет на возможность разложения того или иного выражения. Ниже представлен пример разложения многочленов, имеющих действительные или комплексные корни:

1: "Разложение многочленов (команда Factor)"

$$2: x^4 - 10x^3 + 35x^2 - 50x + 24$$

$$3: (x - 4)(x - 3)(x - 2)(x - 1)$$

4: "В следующем примере используйте опцию Complex"

$$5: x^4 + 9x^3 + 31x^2 + 59x + 60$$

$$6: -(-x - 1 + 2i)(x + 1 + 2i)(x + 3)(x + 4)$$

В первом случае, в строках 2 и 3, числа 1, 2, 3 и 4 являются корнями многочлена #1. Во втором случае многочлен #5 содержит два комплексных корня $(-1-2i)$ и $(-1+2i)$ и два действительных корня (-3) и (-4) . Заметим, что корни многочлена можно непосредственно найти, используя команду solve.

Рассмотрим еще несколько возможностей команды Factor.

1: "Разложение выражений (Factor Trivial)"

2: a SINH (x)

$$3: \frac{a \#e^{-x} (\#e^{2x} - 1)}{2}$$

4: TANH (x)

$$5: \frac{\#e^{2x} - 1}{\#e^{2x} + 1}$$

6: "Разложение выражений (Factor Racional)"

$$7: a^3 - b^3$$

$$8: (a - b) (a^2 + a b + b^2)$$

9: "В этом примере Factor не работает"

$$10: (a - b)^3$$

$$11: (a - b)^3$$

12: "Но можно применить команду Expand #10"

$$13: a^3 - 3 a^2 b + 3 a b^2 - b^3$$

В последнем примере показаны возможности разложения гиперболических функций, выражаемых через экспоненциальные функции. Интересен случай, показанный в строках 9—11. Здесь команда Factor не работает ни при каких установках области факторизации. Однако тут можно использовать команду расширения Expand — результат показан в строке 13.

1.10. Работа со справкой (Help)

Позиция Help основного меню включает систему справки. На экран дисплея выводится перечень ее разделов помощи. Если используется оригинальная (англоязычная) версия системы, то меню помощи выводится в виде:

Derive Help Menu

E — line Editing commands

F — Functions and constants

A — Algebra window commands

2 — 2D—plot window commands

3 — 3D—plot window commands

U — Utility file functions

S — current State of system

R — Return to Derive

Press letter for desired subject

Перевод позиций меню на русский язык дан ниже (в русифицированной Derive меню также имеет этот вид):

E — команды строчного редактора

F — функции и константы

A — команды окна алгебры

2 — окно двумерной графики

3 — окно трехмерной графики

U — функции файлов-утилит

S — текущее состояние системы

R — Возврат в Derive

Нажмите букву, соответствующую выбранному пункту.

В нижней части экрана появляется подменю справки, разделы которой повторяют приведенные выше. Для ввода нужного раздела справки достаточно активизировать соответствующую позицию подменю и нажать клавишу ENTER. На экране дисплея появится текст справки на английском языке.

Организация справки в Derive достаточно проста и очень лаконична. Тем не менее она позволяет получить наиболее важные данные о системе и, главное, о назначении встроенных операторов и функций, а также многих внешних расширений, поставляемых с системой в виде файлов с расширением .mth.

Подсказки можно организовать и при работе с внешними расширениями. Для этого нужно ввести соответствующие комментарии в состав файлов с расширением .mth после знака ; (точка с запятой). Такой комментарий выводится в нижнюю часть экрана. Демонстрационные файлы (с расширением .dmo) используют этот метод. Для его реализации можно использовать любой текстовый редактор, вставив комментарии в нужные места программы.

Ориентация Derive как на символные вычисления, так и числовые, затрудняет анализ всевозможных ошибок, которые могут возникнуть при работе с системой. Поэтому в Derive не предусмотрен подробный вывод всех сообщений об ошибках. Если в каком-либо выражении допущена ошибка или делается попытка выполнить некорректную операцию, система просто повторяет вывод соответствующего выражения (и игнорирует ошибочную операцию) или выводит на экран знак вопроса. Перечень сообщений об ошибках Derive приведен в разделе 1.23.

1.11. Перемещение по строкам документа (Jump)

Как мы убедились, в ходе работы с системой возникает необходимость перемещения по тексту документа, например, с целью поиска необходимых фрагментов выражений и их дальнейшего применения. Для этого используются

клавиши перемещения курсора вверх и вниз. Каждое их нажатие вызывает перемещение на одну строку.

Однако при работе с большими (многострочными) документами такой поиск нужной строки не очень удобен. Поэтому в систему введена команда прямого перехода к указанной строке Jump (прыжок). При ее вводе появляется запрос:

JUMP to:

Надо указать номер строки после знака двоеточия и нажать ENTER. Появится страница документа с указанной строкой.

1.12. Решение уравнений (soLve)

1.12.1. Решение нелинейных уравнений и неравенств

Команда soLve главного меню используется для решения уравнения вида $f(x) = 0$ или в более общем случае $f_1(x) = f_2(x)$. Эти выражения необходимо ввести в текст документа. При решении уравнения вида $f(x) = 0$ достаточно ввести только саму функцию $f(x)$. Тогда при исполнении команды soLve появится запрос:

SOLVE expression: #n

В ответ на него следует указать номер строки, в которой размещено уравнение.

Ниже представлен пример решения трансцендентных тригонометрических уравнений.

```

1: "Решение уравнений - команда soLve"
2: SIN (x) + COS (x) - 0.5
3: x = ATAN (0.142857 SQRT (7)) - 0.25 pi
4: SIN (x) + COS (x) - 0.5 = 0
5: x = ATAN (0.142857 SQRT (7)) - 0.25 pi
6: SIN (x) = - COS (x) + 0.5
7: x = ATAN (0.142857 SQRT (7)) - 0.25 pi
8: "Используя команду arproX находим"
9: x = -0.424031

```

Следует отметить, что команда soLve может использоваться и для решения неравенств. Результатом решения обычно является также неравенство.

1.12.2. Вычисление действительных и комплексных корней полиномов

Команда soLve удобна для решения довольно распространенной (но содержащейся не во всех математических системах) задачи на вычисление корней полиномов. Это иллюстрирует следующий пример.

1: "Вычисление корней полинома - команда solve"

$$2: x^5 + 8x^4 + 31x^3 + 80x^2 + 94x + 20 = 0$$

$$3: x = -2$$

$$4: x = -\sqrt{3} - 2$$

$$5: x = \sqrt{3} - 2$$

$$6: x = -1 - 3i$$

$$7: x = -1 + 3i$$

Как видно, команда solve позволяет найти как действительные, так и комплексные корни полинома. Только надо (см. ниже) установить комплексный формат представления чисел.

1.12.3. Решение неравенств и систем линейных уравнений

Команда solve может использоваться и для решения неравенств и систем уравнений, линейных относительно переменных, для которых ищется решение. Более подробно о технике такого использования команды solve будет рассказано в главах 2 и 4.

1.13. Управление вычислениями и подстановки (Manage)

1.13.1. Управление вычислениями

Многие виды символьных операций Derive выполняет автоматически без какой-либо настройки. Однако в ряде случаев характер преобразований необходимо указывать явно, тем более, что он может быть различным и Derive надо подсказать, в каком направлении ей следует действовать. Для этого служит позиция Manage главного меню. При активизации позиции Manage появляется подменю, содержащее следующие опции.

Branch — расширение по всем действительным корням;

Exponential — экспоненциальное преобразование;

Logarithm — логарифмическое преобразование;

Ordering — явное указание переменных;

Substitute — подстановки в выражения;

Trigonometry — тригонометрические преобразования.

По умолчанию установлена опция Branch. Действие команды Expand при различных опциях Manage (в нашем случае включены тригонометрические преобразования) иллюстрируют следующие примеры:

1: "Действие команды Expand (Manage по умолчанию)"

$$2: \cos(a) \cos(b) - \sin(a) \sin(b)$$

$$3: \cos(b) \cos(a) - \sin(b) \sin(a)$$

- 4: "Преобразование выражения #2 не произошло"
- 5: "Теперь установите Manage Trigonometry Collect"
- 6: $\cos(a + b)$
- 7: "Произошло преобразование #2 в #6"
- 8: "Тригонометрические преобразования для кратных углов"
- 9: "Исходное выражение"
- 10: $\sin(5a)$
- 11: "Используйте Manage Trigonometry Expand Sines"
- 12: $16 \sin^5(a) - 20 \sin^3(a) + 5 \sin(a)$
- 13: "Теперь используйте Manage Trigonometry Collect"
- 14: $\sin(5a)$

Каждое из преобразований, в свою очередь, может содержать опции: Auto (автоматический режим), Collect (собирать) и Expand (расширять).

1.13.2. Подстановки

Одна из самых важных опций в позиции Manage — опция-команда Substitute (подстановка). Она используется для подстановки на место выбранной переменной или фрагмента выражения другого выражения. Следующие примеры показывают применение этой широко распространенной в аналитических преобразованиях операции.

- 1: "Действие команды Substitute"
- 2: $(x - 3)(x - 4)(x - 5)$
- 3: "Командой Substitute x заменено на p+2"
- 4: $(p + 2 - 3)(p + 2 - 4)(p + 2 - 5)$
- 5: "Теперь использована команда Expand"
- 6: $p^3 - 6p^2 + 11p - 6$
- 7: "Для вычисления значения #6 при p=1"
- 8: "Командой Substitute p заменено на число 1"
- 9: "и затем используется команда approx"
- 10: $1^3 - 6 \cdot 1^2 + 11 \cdot 1 - 6$
- 11: 0

Команда `Substitute` может использоваться как для вычислений значений выражений при замене переменных их числовыми значениями, так и для проведения преобразований путем подстановок в символьном виде. Это автоматизирует один из самых распространенных видов математических преобразований — подстановки, которые очень сложно (для больших выражений) выполнять вручную.

1.14. Задание глобальных опций (Options)

Позиция `Options` позволяет задавать ряд глобальных установок (опций) системы, которые действуют при решении любой задачи. Эти опции могут задавать тип дисплея, гамму цветов, точность представления результатов и пр. При активизации позиции `Options` появляется подменю, содержащее ряд команд. Их названия и выполняемые функции описаны ниже.

Color — установка цвета для меню (`Menu`) и рабочей области экрана (`Work`). Работа с этой опцией достаточно очевидна и сводится к установке нужных кодов цветов для различных частей экрана. Коды выводятся в виде ряда цифр от 0 до 15, каждая из которых окрашена соответствующим цветом. Можно выбрать необходимый цвет бордюра и надписей в верхней и нижней части экрана. Опция действует только при работе с цветными видеоадаптерами и дисплеями.

Display — настройка на работу с заданным типом дисплея. Эта опция позволяет установить текстовый или графический режим (`Mode`) работы дисплея, его разрешение (`Reso`), задать установку (`Set`) на стандартный (`Standard`) или расширенный (`Extended`) видеоадаптеры и настроить систему на один из следующих типов видеоадаптеров и дисплеев: `MDA`, `CGA`, `EGA`, `MCGA`, `VGA`, `Hercules`, `AT&T`, `T3100` и `PC jr`.

Execute — прямое исполнение команд `MS-DOS`. Можно, например, просмотреть каталог файлов, выполнить их переименование или удаление и пр. Нажатие затем любой клавиши возвращает систему `Derive` в текущее состояние. Пожалуй, одна из главных возможностей этой команды — возможность просмотра каталога файлов системы с помощью команды `dir`. Например, команда `dir *.mth` позволяет вывести каталог всех файлов системы с расширением `.mth`. Это важно, поскольку сама система `Derive` не выводит каталога файлов, и потому надо помнить имя вводимого файла. Команду `Execute` не следует использовать для ввода резидентных программ (например, `graphics.com`). Такие программы следует вводить до загрузки системы `Derive`.

Input — установка имен переменных и режимов ввода (однобуквенный и многобуквенный режимы, чувствительность к верхнему и нижнему регистрам, режимы перемещения курсора внутри строки ввода).

Mute — установка (по запросам `Yes` или `No`) звуковой сигнализации при обнаружении ошибок.

Notation — выбор одной из следующих нотаций (представлений) чисел: десятичной (`Decimal`), смешанной (`Mixed`), рациональной (`Rational`), научной (`Scientific`). Кроме того, задается число знаков результата (`Digit`), по умолчанию оно равно 6. Ниже даны примеры на изменение нотаций численных данных.

- 1: "Различные формы представления чисел"
- 2: "Исходное выражение"

- 3:
$$\frac{1000000}{3} x$$
- 4: "Опция Notation Decimal"
- 5: 333333.3 x
- 6: "Опция Notation Mixed"
- 7: 333333.3 x
- 8: "Опция Notation Racional"
- 9:
$$\frac{1000000}{3} x$$
- 10: "Опция Notation Scientific"
- 11:
$$3.33333 \cdot 10^5 x$$

Precision — установка точности вычислений. Возможна установка следующих режимов: приближенные вычисления — аппроксимация (Approximate), точные целочисленные вычисления (Exact) и смешанные (Mixed). Здесь также можно задать число верных знаков результата (Digit). Приведенные ниже примеры показывают, как можно изменять точность вычислений.

- 1: "Действие команды Precision"
- 2: "Исходное выражение:"
- 3:
$$\frac{40!}{3}$$
- 4: "Установлена опция Approximate Digit:6"
- 5: 2.719717612088184295268763933614069853851870200644 10⁴⁷
- 6: "Установлена опция Exact Digit:50"
- 7: 2.7197176108263257811520375653203863142399999999999 10⁴⁷
- 8: "Установлены опции Mixed и Decimal (Digit:50)"
- 9: 271971761082632578115203756532038631424000000000
- 10: "Новое выражение:"
- 11:
$$\frac{1}{7}$$
- 12: 0.14285714285714285714285714285714285714285714

Radix — установка основания для числовых данных в пределах от 2 до 36, по умолчанию — 10. Основания могут устанавливаться отдельно для входных данных (Input) и выходных (Output). Это позволяет легко решать задачи на преобразование чисел с одним основанием в числа с другим основанием.

- 1: "Перевод двоичных чисел в десятичные"
- 2: "Radix Input:2 Output:10"
- 3: "Вводим двоичное число 11001110 и получаем"
- 4: 0 CE
- 5: "Перевод десятичных чисел в шестнадцатеричные"
- 6: "Radix Input:10 Output:16"
- 7: "Вводим десятичное число 10000 и получаем"
- 8: 2710
- 9: "Вводим десятичное число 65000 и получаем"
- 10: 0 FDE8

Последний пример показывает, что Derive может работать не только с десятичными числами, но и с числами, имеющими различные основания. Это двоичные, восьмеричные и шестнадцатеричные числа, широко используемые в вычислительной технике. Разряд шестнадцатеричных чисел представлен цифрами от 0 до 9 и затем латинскими буквами A, B, C, D, E и F.

1.15. Выход из системы (Quit)

Если необходимо временно выйти из системы Derive в MS-DOS, используется команда Execute из позиции основного меню Options. При этом в строке ввода можно задать исполнение какой-либо команды MS-DOS, например вывести каталог файлов или уничтожить ставший ненужным файл. По окончании действия команды MS-DOS система возвращается к предшествующему этой команде состоянию. Все данные и определения сохраняются.

Команда Quit основного меню используется, если необходимо окончательно покинуть систему и вернуться в MS-DOS или в среду администратора диска (Norton Commander, например). Перед тем как войти в MS-DOS, система дает запрос:

Abandon expression (Y/N)? <Покинуть условие (Да/Нет)?>

Ответ Y приведет к прекращению сеанса работы с системой. При этом текст текущего документа и все данные и определения будут утрачены. Поэтому, если они представляют ценность, следует позаботиться об их сохранении (см. описание команд позиции Transfer основного меню). Тогда следует ответить на запрос N, использовать команду Save в позиции основного меню Transfer и уже затем выйти из системы.

1.16. Работа с графикой (Plot)

1.16.1. Построение двумерных графиков в декартовой системе координат

Derive позволяет строить графики различных функций с автоматической установкой масштабов, обеспечивающих большие размеры графиков и предельно высокое их разрешение. Возможно построение графиков следующих типов: функций одной переменной (двумерная 2D-графика), параметрически заданных функций, графиков функций в полярной системе координат и графиков функций трех переменных (трехмерная 3D-графика).

Derive автоматически анализирует вид функции, для которой используется команда Plot. Если это функция одной переменной вида $y = f(x)$, то задается режим построения 2D-графики. Если задана функция в параметрическом виде $[x(t), y(t)]$, то будет построен ее график. Задание функции в полярном виде $[f(\theta)]$ позволяет построить график функции в полярной системе координат, а для функции $[f(x, y)]$ строится график трехмерной поверхности.

При работе с двумерной графикой Derive позволяет на одном экране строить несколько графиков для различных функций. Если это допускают возможности дисплея, графики разных функций выделяются разными цветами. Derive строит графики в два захода. Вначале редкими точками быстро выполняется грубое построение графика, а затем уже точки заменяются линией. Это полезно, поскольку некоторые типы графиков требуют большого времени построения. Если пользователь из грубого хода графика видит, что некоторые параметры (например, масштабы) его не удовлетворяют, он может преждевременно прервать построения, нажав Esc или на клавишу с заглавной буквой команды. О построении графика говорит надпись в третьей строке нижнего окна. Построения заканчиваются, когда она сменяется надписью Enter option (введите опцию).

В поле графиков двумерных функций расположен графический курсор — метка в виде небольшого крестика. Координаты курсора непрерывно отображаются, а сам курсор может плавно перемещаться в любое место. Обычно он используется для отметки характерных точек графика, координаты которых затем легко определяются. Например, можно отметить курсором точку пересечения двух кривых, тогда его координаты определяют решение системы из двух уравнений, описывающих эти кривые. Удобно так определять и характерные точки функций, например их экстремумы.

Выбор типа графика задается типом функции, для которой предполагается строить график. Рассмотрим вначале работу с 2D-графикой. Для этого вводится команда Plot, создающая подменю выбора со следующими позициями:

Beside **Under** **Overlay**
Рядом **Ниже** **С перекрытием**

По умолчанию задана позиция Beside. Тогда графическое окно открывается рядом с окном формул — достаточно указать лишь границу раздела. После появления графического окна внизу его появляется меню графики со следующими позициями:

Algebra — обеспечивает возврат к главному меню и заданию функции;
Center — изменяет положения центра координатной системы (он занимает такое положение, при котором графический курсор оказывается в центре графика);

- Delete** — стирает график или графики (команда имеет опции: All — стирает всех графиков, Butlast — стирает предыдущий график, First — стирает последний график);
- Help** — включает справочную систему;
- Move** — быстро устанавливает курсор по заданным его координатам;
- Options** — устанавливает опции графика (Accuracy — точность построения графика, Color — установка цветов, Display — установка типа дисплея, Execute — прямое исполнение команд MS-DOS, Mute — включение звукового контроля при ошибках, Notation и Precision — установка нотации и точности представления чисел, Radix — установка масштабов графиков и State — установка типа координатной системы Rectangular и Polar, т. е. декартова прямоугольная или полярная);
- Plot** — обеспечивает построение очередного графика;
- Quit** — обеспечивает выход из Derive в MS-DOS с прекращением работы;
- Scale** — изменяет масштаб (одного деления масштабной сетки из точек);
- Ticks** — изменяет густоту масштабной сетки (с параметрами Rows — строки и Columns — столбцы, параметры задаются в знакахместах);
- Windows** — управляет окнами;
- Zoom** — изменяет размеры окна в два раза по обеим осям координат (опция Both) или по осям X и Y (опции X, Y), уменьшает или увеличивает (по запросу Direction: In Out).

Для управления функциями двумерной графики можно использовать быстрые команды, вводимые в основном функциональными клавишами (в позиции Windows основного меню):

- F1** — переключить на следующее окно;
- F2** — переключить перекрывающиеся окна;
- F5** — переключить на предшествующий режим дисплея;
- F7** — увеличить размер графика по оси y;
- F8** — уменьшить размер графика по оси y;
- Shift + F7** — увеличить размер графика по оси x;
- Shift + F8** — уменьшить размер графика по оси x;
- F9** — увеличить размер графика по обеим осям;
- F10** — уменьшить размер графика по обеим осям;
- Shift + F10** — напечатать весь экран;
- Shift + F9** — напечатать текущее окно;
- Ctrl + F10** — послать весь экран в TIF-файл;
- Ctrl + F9** — послать текущее окно в TIF-файл.

Заметим, что возможность записи изображений в TIF-файл появилась лишь у версии Derive 2.60. У предшествующих версий такой возможности не было. В версии 2.60 появилась и возможность печати лишь содержимого текущего окна, а не всего экрана. Это часто полезно, так как при копировании всего экрана внизу отображается, как правило, ненужное главное меню.

С помощью позиции Options можно вызвать меню, содержащее следующие операции:

Accuracy (Точность)	Color (Notation)	Display (Дисплей)	Execute (Выполнение)	
Mute (Сигнал)	Notation (Нотация)	Precision (Точность)	Radix (Основание)	State (Тип)

Позиция Assigasy задает количество отрезков линий, из которых строится график. Чем их больше, тем меньше заметна дискретность построения графиков, но больше время их построения. Опция Color позволяет установить для каждой кривой графика цвет, который выбирается из меню в виде кодов цветов, цифры которых окрашены соответствующим цветом. При этом можно управлять цветами графиков (Plot), меню (Menu) и рабочей частью экрана (Work). Опция Display настраивает систему на работу с соответствующим дисплеем, а опция State позволяет установить тип координатной системы: декартова прямоугольная (Rectangular) или полярная (Polar). Остальные опции соответствуют описанным для режима вычислений Algebra.

Рассмотрим построение графиков двух функций в декартовой системе координат. Зададим (с помощью позиции Authors главного меню) две функции — $\sin(x)/x$ и $\tan(x)$:

1: "Построение графика функции"

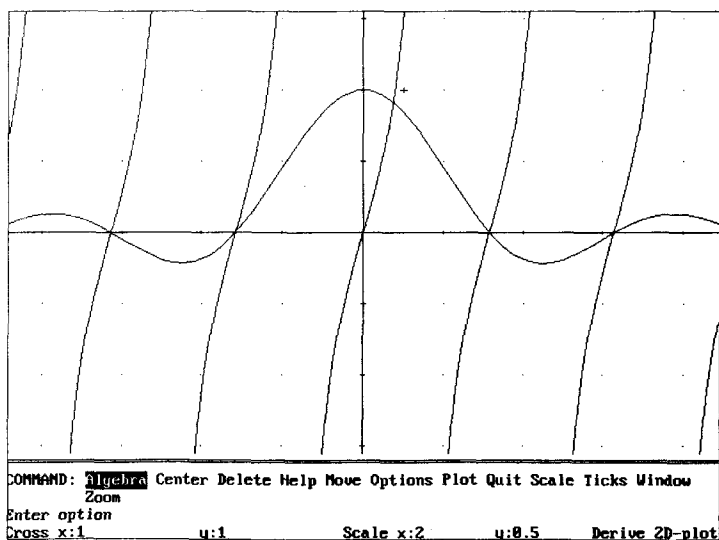
2:
$$\frac{\text{SIN}(x)}{x}$$

3: $\text{TAN}(x)$

Далее, используя команду Plot и установив Scale $y = 2$ и $x = 2$, получим графики этих функций (рис. 1.7).

Применив различные команды центрирования и масштабирования графика, можно представить его в любом удобном виде. Графики выглядят достаточно просто, но вполне адекватно изображению функции.

Рис. 1.7
Графики
двух функций
в декартовой
системе
координат



1.16.2. Построение графика параметрически заданных функций

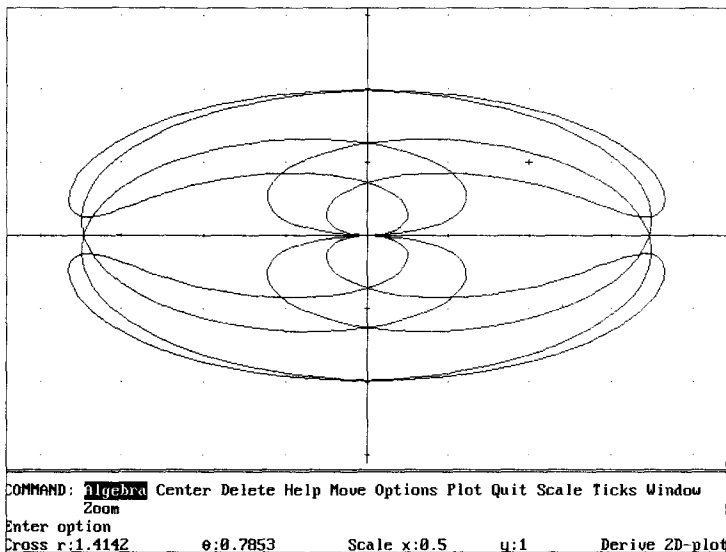
Теперь рассмотрим построение графика параметрически заданных функций на примере построения фигуры Лиссажу.

Очистим окно (команды Transfer и Clear). Затем введем следующее выражение, описывающее фигуру Лиссажу в параметрической форме:

$$[2*\sin(2*t), 3*\cos(3*t)]$$

Используя команду Plot, получим график, показанный на рис. 1.8.

Рис. 1.8
График
параметрически
заданной
функции



1.16.3. Построение графика функций в полярной системе координат

Derive позволяет строить графики функций в полярной системе координат. Для этого надо установить тип координатной системы Polar в опции Radix и задать функцию в виде:

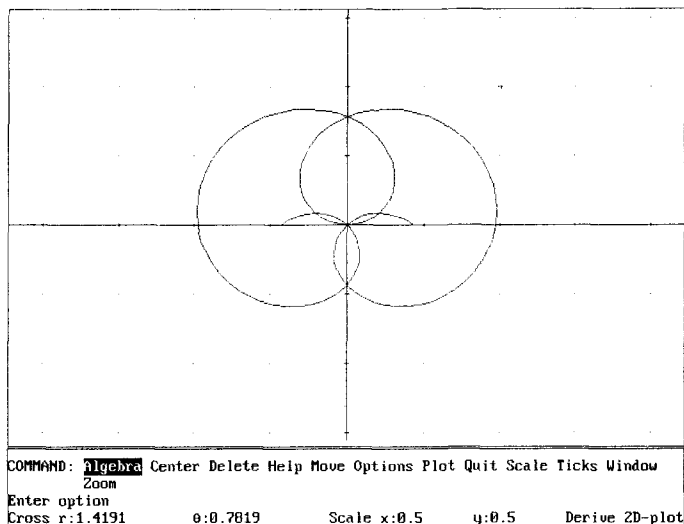
$$r = f(\text{theta})$$

Пусть нужно построить график функции

$$r = \sin(t*4/7)$$

Достаточно ввести правую часть этого выражения и использовать команду Plot. После перехода в графическое окно надо, используя команду Polar опции State, установить полярную систему координат. Далее, указав команду Plot, можно построить график функции, показанный на рис. 1.9.

Рис. 1.9
График функции
в полярной
системе
координат



1.16.4. Построение графиков функций в трехмерном пространстве

Система Derive содержит развитые средства для построения графиков поверхностей, описываемых уравнениями вида $z = f(x, y)$, т. е. трехмерных (или 3D-типа) графиков. Предусмотрена возможность автоматического масштабирования графиков и удаления невидимых линий. Последнее делает графики более наглядными и эстетичными.

Для построения графика вначале необходимо (команда AutoZ основного меню) ввести выражение для $f(x, y)$, после чего использовать команду Plot. Derive автоматически анализирует записанную функцию, и если она является функцией двух переменных, команда Plot включает режим построения 3D-графиков. Она имеет подменю следующего вида:

Algebra **Center** **Eye** **Focal** **Grids** **Hide** **Lenght**
 <Алгебра> <Центр> <Взгляд> <Фокус> <Сетка> <Скрыть> <Длина>

Более подробное назначение команд подменю следующее:

Algebra — переключение в окно Algebra;
Center — установка координат центра фигуры;
Eye — установка координат обзора фигуры;
Focal — установка координат фокальной точки;
Grids — установка числа линий каркаса фигуры;
Hide — включение/выключение невидимых линий;
Length — установка длины стороны "ящика".
Option — установка опций графики;
Plot — включение построения графика;
Quit — выход из системы Derive;
Window — работа с окнами;
Zoom — установка масштабов графиков.

Четыре отмеченные опции (Center, Eye, Focal и Lenght) вызывают подменю:

Имя опции: x:n1 y:n2 z:n3 Auto:(Yes) No

Их параметрами являются размеры или координаты в виде чисел n_1 , n_2 и n_3 . После переустановки параметров следует повторить команду Plot для построения графика.

Опция Grid вызывает подменю:

GRIDS: x:n1 y:n2

Она устанавливает число линий каркаса, в виде которого строится фигура, по осям x и y . Чем выше значения n_1 и n_2 , тем лучше прорабатываются детали фигуры, но построение идет дольше. Оптимальные значения n_1 и n_2 можно уточнить лишь экспериментально (по умолчанию они равны 10).

Опция Hide также содержит подменю:

HIDE: Lines: Yes No

В зависимости от установки будет включен или выключен алгоритм устранения невидимых линий. Если он выключен, фигура напоминает каркас из тонких проволочек, висящий в пространстве.

С помощью опций позиции Options (в основном их назначение аналогично описанному для 2D-графики) можно управлять точностью построений и цветовой гаммой трехмерной графики. Тут предусмотрена возможность задания различного цвета для верхней и нижней видимых поверхностей трехмерных фигур, что делает их более наглядными.

Для управления функциями окон в режиме построения 3D-графиков можно использовать быстрые команды, вводимые функциональными клавишами:

- F1** — перейти к следующему окну;
- F2** — переключить перекрывающиеся окна;
- F5** — переключиться на предыдущий режим дисплея;
- F7** — увеличить размер графика по оси y ;
- F8** — уменьшить размер графика по оси y ;
- Shift + F7** — увеличить размер графика по оси x ;
- Shift + F8** — уменьшить размер графика по оси x ;
- F9** — увеличить размер графика по обеим осям;
- F10** — уменьшить размер графика по обеим осям;
- Shift + F10** — напечатать весь экран;
- Shift + F9** — напечатать текущее окно;
- Ctrl + F10** — послать весь экран в TIF-файл;
- Ctrl + F9** — послать текущее окно в TIF-файл.

Рассмотрим приведенный ниже пример.

Пусть нужно построить график следующей функции двух переменных:

$$z(x, y) = \cos(0.04 * (x^2 + y^2)) \quad (1.1)$$

Введем правую часть выражения, используя команду Author, а затем используем команду Plot. Далее, используя команду Length, установим масштабы по осям x , y и z соответственно равными 15, 15 и 2, а с помощью команды Grids зададим по 20 линий каркаса по осям x и y . Далее, используя команду Plot, получим график, показанный на рис. 1.10.

Теперь введем новую функцию в виде:

$$\cos((x^2 + y^2)/4) / (3 + x^2 + y^2) \quad (1.2)$$

Рис. 1.10
График
трехмерной
поверхности
(1.1)

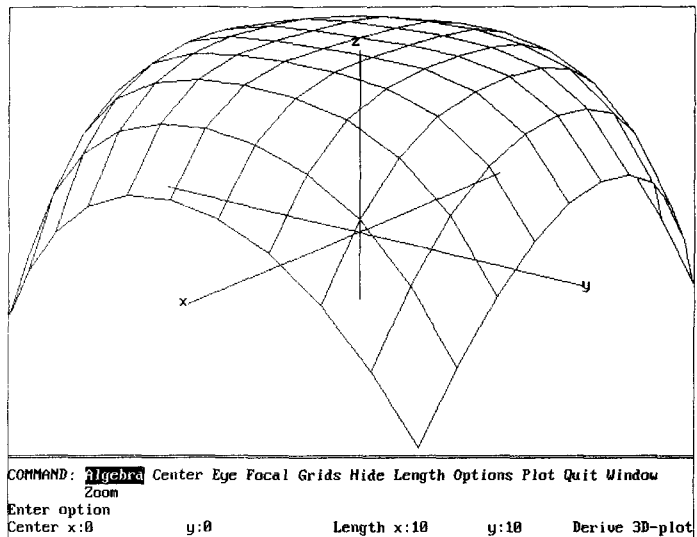


График этой функции (рис. 1.11) напоминает шляпу с более острой вершиной, чем у фигуры, показанной на рис. 1.9.

Далее построим график функции:

$$-\sqrt{\text{abs}(x)} - \sqrt{\text{abs}(y)} \quad (1.3)$$

Этот график показан на рис. 1.12.

И, наконец, введем еще одну функцию:

$$\sqrt{\text{abs}(x)} - \sqrt{\text{abs}(y)} \quad (1.4)$$

Ее график, показанный на рис. 1.13, наглядно иллюстрирует, как резко меняется изображение даже при смене знака у одного из членов выражения —

Рис. 1.11
График
трехмерной
поверхности
(1.2)

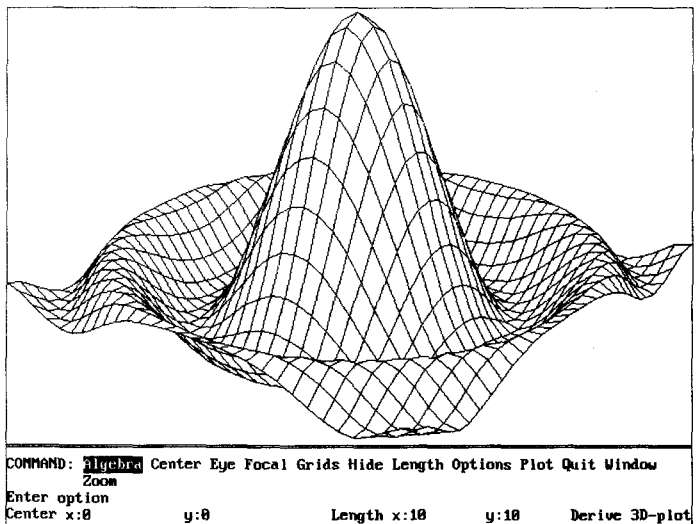
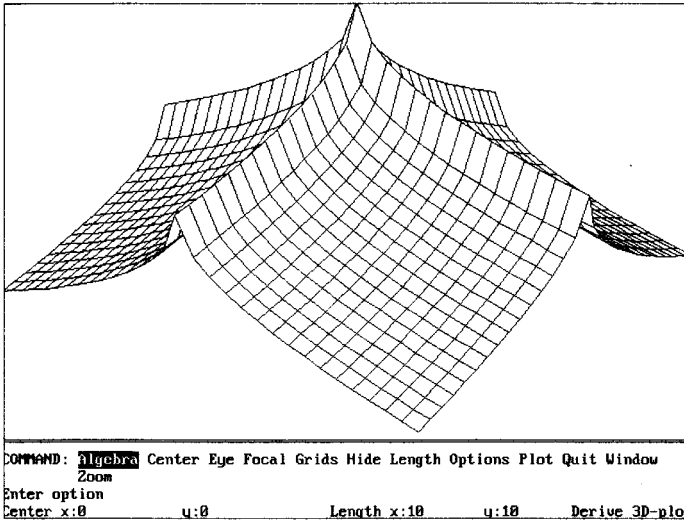


Рис. 1.12

График
трехмерной по-
верхности (1.3)



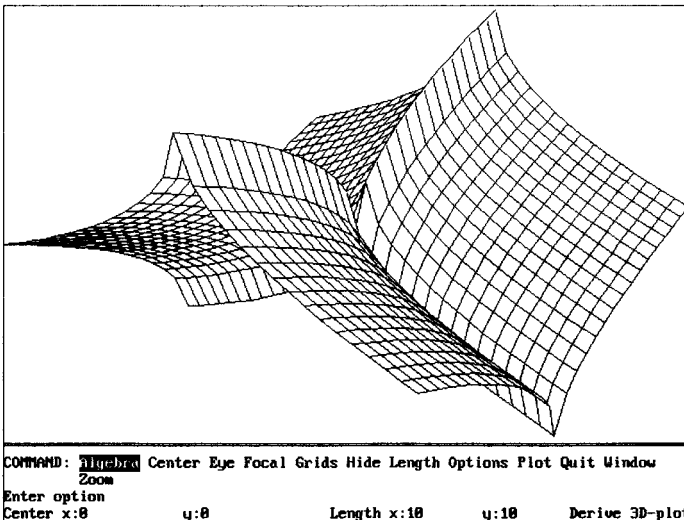
только этим и отличается выражение (1.4) от (1.3). Фактически поверхность (1.4) обращена пиком вниз, тогда как (1.3) — пиком вверх.

На рис. 1.10 видны координатные оси x , y и z , которые система строит по умолчанию. Однако с помощью опции Axis их можно убрать, с тем чтобы они не искажали вид рисунка (где оси оставлять, а где убирать — это дело пользователя). На рис. 1.11–1.13 оси убраны. Рис. 1.10–1.11 наглядно иллюстрируют богатые возможности математической трехмерной графики Derive. Большое число примеров задания трехмерных фигур дается в файле plot3d.mth. Их рекомендуется самостоятельно проработать.

К сожалению, Derive не предусматривает запись графиков на диск. Однако для этого можно использовать соответствующие резидентные программы из других графических пакетов. После загрузки такой резидентной програм-

Рис. 1.13

График
трехмерной
функции (1.4)



мы, например prtscr.com, копию графика можно распечатать на принтере, нажав одновременно клавиши Shift и Print Screen (на некоторых ПК только Print Screen).

1.17. Уничтожение строк (Remove)

В ходе диалога с Derive каждое вводимое командой Author выражение и каждое действие по его преобразованию или вычислению отображается в очередной пронумерованной строке. В результате текст документа быстро засоряется ненужными записями, так как, прежде чем получить правильный (или нужный) результат, пользователь проводит многочисленные промежуточные преобразования.

Для удаления строк, содержащих ненужную информацию, служит команда уничтожения строк Remove. При ее активизации система выводит в нижней части экрана запрос:

REMOVE: Start:n1 End:n2

По умолчанию здесь указан начальный n1 и конечный n2 номер в виде номера последней введенной строки. Если нажать клавишу Enter, то эта строка будет уничтожена. Можно, однако, указать любые номера существующих строк и стереть группу строк от строки с начальным номером до строки с конечным номером.

Следует помнить, что стирание строк не означает полное уничтожение результатов операций, которые в них выполнялись. Напротив, система сохраняет эти операции (например, определенные переменные, векторы, матрицы, функции пользователя и т. д.) в памяти и их можно использовать в дальнейшем. Это справедливо, даже если уничтожаются все строки.

Удаление ставших ненужными строк и ввод новых заполняют память компьютера. Следует также учитывать, что многие операции требуют временного привлечения значительных объемов памяти. В нижней части экрана содержится сообщение вида Memoгу pp% о том, сколько процентов памяти компьютера занято. Надо следить, чтобы значения pp% не приближались к 0, так как в этом случае работа системы будет прекращена с выдачей сообщения о переполнении памяти.

Для полной очистки системы без выхода из нее следует использовать команду Clear в позиции Transfer основного меню.

1.18. Упрощение выражений (Simplify)

Команда Simplify — одна из самых важных команд системы Derive. В общем случае она выполняет упрощение символьных выражений (см. в главе 4 более подробное пояснение функций этой команды). При операциях с числовыми выражениями команда Simplify позволяет представить результат вычислений в виде рациональных чисел. Ниже даются примеры применения команды Simplify.

1: "Работа с командой Simplify"

$$2: \frac{2}{5} + \frac{4}{10} + \frac{45}{100}$$

$$3: \frac{5}{4}$$

$$4: \frac{a}{b c x} + \frac{a + b}{(b c + b^3)(x - b)} - \frac{(c - a b) x + (b + a) c}{(c^2 + b^2 c)(x^2 + c)}$$

$$5: \frac{x + a}{x(x^2 + c)(x - b)}$$

$$6: (x + a) \sin^2(x) + (x + a) \cos^2(x) - (x + a) + 1$$

$$7: 1$$

Примеры в строках 4–7 наглядно показывают, насколько сильно могут упрощаться алгебраические и тригонометрические выражения до и после выполнения команды, разумеется, если такие упрощения принципиально возможны. Не следует полагать, что упрощенные выражения обязательно совпадут с теми, что даются в справочниках; это произойдет только в том случае, когда результат упрощений однозначен или когда путь к нему квалифицированно подсказывается пользователем.

1.19. Работа с внешними устройствами (Transfer)

Для работы с дисковыми накопителями и принтером служит команда `Transfer`. С помощью этой команды и соответствующих подменю система позволяет записывать документы на диск и считывать их с диска, а также посылать их на распечатку принтером или в дисковый файл в особом (принтерном) формате представления.

Команда `Transfer` вызывает подменю, имеющее ряд позиций. Они и относящиеся к ним действия, приведены ниже.

Load — загрузка нового файла с расширением `.mth` с уничтожением текущего файла;

Save — запись файлов на диск в различном формате;

Merge — считывание с диска файла и подключение его к текущему файлу без очистки памяти;

Clear — полная очистка памяти с уничтожением текущего документа и относящихся к нему данных;

Demo — загрузка с диска демонстрационных файлов с расширением `.dmo` и запуск их на исполнение;

Print — создание файлов для распечатки документов и направление их на принтер или диск.

Ниже перечисленные команды рассмотрены более подробно.

1.19.1. Работа с дисковыми накопителями

При подаче команды Load появляется дополнительное подменю:

TRANSFER LOAD:	Derive	State	Utility
Загрузка	Derive	Статус	Утилиты

Это подменю определяет тип загружаемого с диска файла. В позиции Derive загружаются тексты документов — файлы с расширением .mth. Они по мере загрузки выводятся на экран дисплея. Команда State загружает файл derive.ini, хранящий данные о состоянии системы. Наконец, команда Utility также загружает файлы с расширением .mth с внешними расширениями (утилитами) системы. Однако при этом загружаемые документы на экран дисплея не выводятся. Во всех случаях система запрашивает имя файла. По команде Merge загружаются лишь файлы с расширением .mth. Отличительная черта команды Demo — автоматический запуск демонстрационного файла.

Файлы с расширением .mth являются текстовыми файлами, причем набор символов в них совпадает с набираемыми на клавиатуре и отображаемыми в нижней части экрана знаками. Номера строк в этом файле отсутствуют. Ниже дана распечатка файла, иллюстрирующая это.

```
"Зададим выражение (команда Author):"
SIN(x)^2+COS(x)^2
"Вычисляем #2, используя команду Simplify"
1
"Выполним разложение на простые дроби выражения"
(x^2-5)/(x*(x-1)^4)
"Разложим #6 на простые дроби (команда Expand):"
-4/(x-1)^4+6/(x-1)^3-5/(x-1)^2+5/(x-1)-5/x
"Для выражения #8 используем команду Simplify:"
(5*x^3-20*x^2+31*x-20)/(x-1)^4-5/x
"Для #8 используем команду Factor Racional:"
(x^2-5)/(x*(x-1)^4)
"Для #10 используем команду Factor Racional:"
(x^2-5)/(x*(x-1)^4)
```

Файл derive.ini также является текстовым файлом и содержит многочисленные данные о состоянии системы во время записи этого файла:

```
*PRECISION* |Exact|
*PRECISION-DIGITS* 6
*NOTATION* |Rational|
*NOTATION-DIGITS* 6
*INPUT-BASE* 10
```

```
*OUTPUT-BASE* 10
*INPUT-MODE* |Character|
*CASE-MODE* |Insensitive|
*ARROW-KEY-MODE* |Line-edit|
*BRANCH* |Principal|
*EXP-EXPD* |Auto|
*LOG-EXPD* |Auto|
*TRIG-EXPD* |Auto|
*TRIG-SQR* |Auto|
*ROWS/TICK* 4
*COLS/TICK* 8
*PLOT-ACCURACY* 7
*COORDINATE-SYSTEM* |Rectangular|
*PLOT-POINTS* |Discrete|
*POINT-SIZE* |Large|
*AUTO-COLOR* |Yes|
*AXES-COLOR* 7
*CROSS-COLOR* 15
*TOP-COLOR* 15
*BOTTOM-COLOR* 7
*VIDEO-MODE* 18
*PREVIOUS-MODE* 3
*CHARACTER-SET* |Extended|
*PREVIOUS-SET* |Extended|
*ORDER* (-1 \x \y \z)
*DEFAULT-INTV* (((((1 INF) . -1) . T) (INF) . T)
*MUTE* |No|
*INSERT-MODE* NIL
*SAVE-LENGTH* 79
*PAGE-LINES* 66
*PAGE-COLUMNS* 80
*TOP-MARGIN* 0
*BOTTOM-MARGIN* 0
*LEFT-MARGIN* 8
*RIGHT-MARGIN* 3
*PRINT-STYLE* |Standard|
*PRINTER-TYPE* |Plain|
*PRINT-ORIENTATION* |Portrait|
*PRINT-SIZE* |Large|
*PAPER-SIZE* |Standard|
*PRINTER-INIT* ||
*PRINT-BACKGROUND* |White|
*PRINT-DEVICE* PRN
*PCL-SMALL-FONT* |←(10U←(s0p16.6h8.5v0s0b0T←&14.8C←*p0x0Y|
*PCL-LARGE-FONT* |←(10U←(s0p10h12v0s0b3T←&18D←*p0x0Y|
*FRAME-COLOR* 7
*OPTION-COLOR* 7
```



```
*PROMPT-COLOR* 15
*STATUS-COLOR* 7
*MENU-BACKGROUND* 0
*BORDER-COLOR* 0
*WORK-COLOR* 15
*WORK-BACKGROUND* 0
```

Заметим, что размеры распечаток документов в этом файле даются в знакоместах, а цвета заданы их стандартными кодами. Такие файлы формируются самой системой, но их можно скорректировать и с помощью внешнего текстового редактора.

При работе с командой Save появляется подменю:

TRANSFER SAVE: Derive Basic Fortran Pascal Options State

Первые четыре команды обеспечивают запись документов (по запросу их имени) в виде файла с форматом Derive, Basic, Fortran и Pascal. Это позволяет вводить документы в программы, написанные на наиболее распространенных языках высокого уровня. Команда Options позволяет записать опции системы (она имеет подменю Range: All Some Length). Наконец, команда State позволяет записать текущее состояние системы (по умолчанию в файл derive.ini).

1.19.2. Работа с принтером

Пункт основного меню Print содержит подменю:

TRANSFER PRINT: Printer File Layout Options

Команда Printer обеспечивает распечатку текущего документа принтером, тогда как команда File (с запросом имени файла) направляет текст документа в файл с расширением .prt. Вид распечаток этого файла (что видно из многочисленных примеров) отличается от вида распечаток файлов с расширением .mth. Файлы с расширением .prt создают изображения математических формул, приближенные к естественным и тем, которые наблюдаются в верхней части экрана дисплея.

Команда Printer содержит подменю с тремя командами:

Expression — печать математических выражений;
Screen — печать копии экрана;
Windows — печать содержимого окна (All — всех окон, Current — только активного окна).

Аналогичные команды содержит и команда File — в этом случае данные направляются в файл. Команда Layout позволяет установить размер страницы и полей:

```
Length:66 Width:80 Top:0 Bottom:0 Left:8 Right:3
(Длина) (Ширина) (Верх) (Низ) (Слева) (Справа)
```

Наконец, команда Options устанавливает диапазон печатаемых выражений, набор символов, тип принтера и ориентацию и размер страницы. Эти опции определяют (в знакоместах) длину и ширину печати, а также размеры полей сверху, внизу, слева и справа.

1.20. Перемещение строк (moVe)

В процессе подготовки документов часто возникает необходимость перемещения различных частей документов. Поскольку документ состоит из пронумерованных строк, то для этого достаточно иметь возможность перемещать одну или несколько строк и вставлять их между строками текущего документа.

Эта возможность в Derive реализуется с помощью команды moVe основного меню. При ее использовании система дает запрос:

MOVE: **Before:n1** **Start:n2** **End:n3** **Enter label number:**
(Перенести) **(Перед)** **(Начало)** **(Конец)** **(Введите номер строки)**

После ввода чисел n1, n2 и n3 строки, начиная с n2 и кончая n3, будут перенесены на место перед строкой n1. Если перед строкой n1 нужно вставить только одну строку, то нужно ее номер указать как n2 = n3.

Команда moVe не несет каких-либо иных функций, кроме функции редактирования текстов документов. Она обычно используется совместно с командой Remove, позволяющей уничтожить заданный блок строк. С помощью этих команд можно навести должный порядок в документе перед его записью на диск.

1.21. Работа с окнами (Window)

Система Derive позволяет разбить экран дисплея на несколько окон и вывести в каждое из них свою текстовую и графическую информацию. Для проведения с окнами различных операций используется позиция основного меню Window. Она содержит подменю со следующими командами:

Close — закрыть активное окно;
Designate — задать тип окна (Algebra, 2D или 3D Plot);
Flip — стереть окно;
Go to — перейти в окно с указанным номером;
Next — перейти к следующему окну;
Previous — перейти к предыдущему окну;
Open — открыть окно;
Split — разделить окно на две части.

Действие большинства этих команд достаточно очевидно.

1.21.1. Разбиение окон

Команда Split используется для деления активного окна на две части — по горизонтали или по вертикали. При этом выводится запрос:

WINDOW SPLIT: **Horizontal** **Vertical**
(Деление окна) **(По горизонтали)** **(По вертикали)**

Затем задается вопрос о том, по какой строке или по какому столбцу производится деление. После разбивки окна нужно определить, что именно будет выводиться в каждое новое окно. Для этого используется команда Designate. Она вызывает подменю с запросами:

WINDOW DESIGNATE: Type: Algebra 2D-plot 3D-plot

После ответа на эти очевидные запросы последующие действия выполняются так, как при уже описанной выше работе с текстами документов и двумерной и трехмерной графикой.

1.21.2. Управление окнами

Для управления окном с текстами документов используются следующие клавиши:

F1	— перейти к следующему окну;
F2	— переключить перекрывающиеся окна;
F3	— вставить выделенное выражение;
F4	— вставить выделенное выражение, заключив его в скобки;
F5	— переключиться на предыдущий режим дисплея;
Shift + F10	— напечатать весь экран;
Shift + F9	— напечатать текущее окно;
Ctrl + F10	— послать весь экран в TIF-файл;
Ctrl + F9	— послать текущее окно в TIF-файл.

1.21.3. Примеры работы с окнами

Используя команды работы с окнами, можно на одном графике представить запись арифметических выражений и графики различных функций. Покажем это на примере.

Пусть нужно построить одновременно графики четырех функций одной переменной x , расположив их и соответствующие расчетные выражения в восьми окнах. Для этого:

1. Используя команды Window и Split, разделим экран по вертикали на два полуокна. Затем, используя те же команды, разделим их по горизонтали еще на четыре окна.

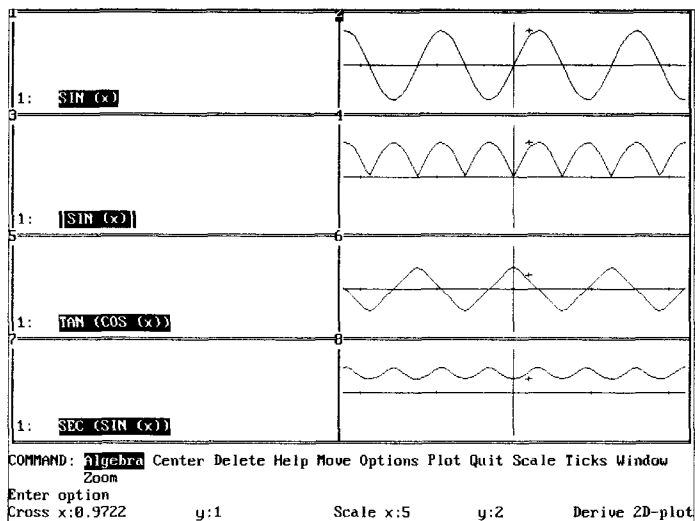
2. Используя команду Window и Goto 1, перейдем в окно 1 (номер окна виден в его верхнем левом углу, активное окно помечено другим цветом). Введем в это окно первое выражение (см. рис. 1.9), используя команду Author.

3. Используя команду Window Next, перейдем в следующее окно — 2. Далее с помощью команд Window Designate 2D Plot установим режим построения графика. Далее, как было описано, зададим построение графика функции, установив желаемые масштабы.

4. Аналогично зададим еще три функции и построение еще трех графиков. Окончательный вид изображения на экране дисплея показан на рис. 1.14.

Рис. 1.14 показывает, что использование многооконного режима существенно повышает наглядность представления как формульной, так и графической информации. Появляется возможность наглядно сопоставлять формы разных графических зависимостей, многие из которых весьма полезны для моделирования различных сигналов.

Рис. 1.14
Графики
четырёх
функций

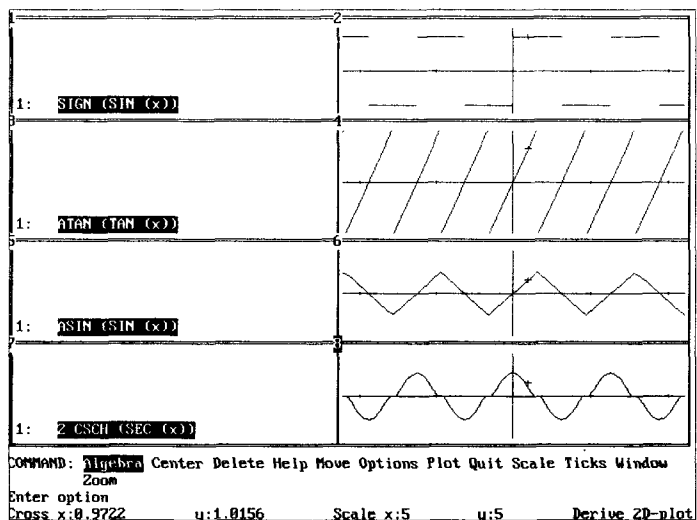


На рис. 1.15 показаны формулы и графики других четырех периодических функций — сглаженного треугольного сигнала, треугольного, пилообразного и прямоугольного сигналов. Хотя используемые в этих простых выражениях элементарные функции общеизвестны, для многих (включая и автора) откровением является, что их комбинации дают такие (широко применяемые в электро- и радиотехнике) сигналы.

Derive позволяет выводить в окна графики разных типов, например 2D в декартовой и полярной системах координат и 3D-графики. Это иллюстрирует следующий пример.

Пусть требуется создать четыре окна: одно текстовое с записью трех функций и три с графиками функций. Будем действовать следующим образом:

Рис. 1.15
Графики
других
четырёх
функций

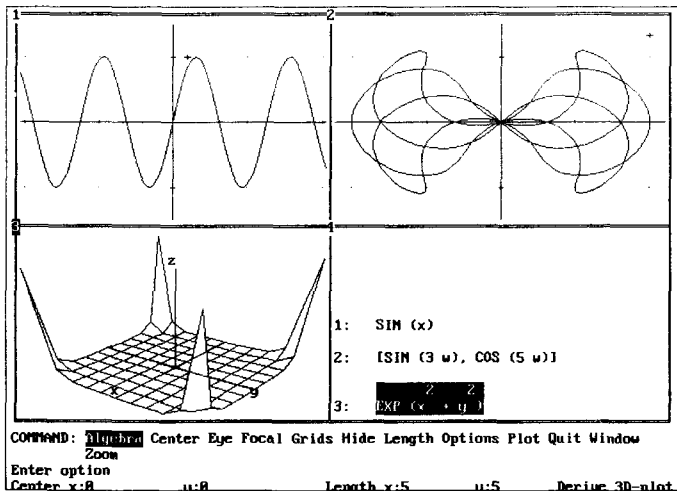


1. Введем с помощью команды Author три выражения:
 $\sin(x)$ — синусоида;
 $[\sin(3*w), \cos(5*w)]$ — параметрическая функция;
 $\exp(x^2+t^2)$ — функция двух переменных.
2. Разделим окно на два окна. Для этого используем команды Window Split и Vertical (напоминаем, что тут достаточно нажать клавиши W, S и V). Деление окон соответствует текстовому формату экрана, т. е. окно состоит из 80 столбцов. Так что, согласившись с разделом окна по 40 столбцу, получим на экране два окна — левое и правое. Затем разделим их надвое по горизонтали. Указанные выражения окажутся во всех окнах.
3. Используя команды Window и Goto 1, сделаем окно 1 активным. Выделим в нем выражение #1. Затем, используя команды Window и Designate, перепределим окно в графическое и (как было описано раньше) построим в нем график синусоиды.
4. Используя команду Window Next, перейдем в окно 2. Выделим в нем выражение #2. Затем, используя команды Window Designate 2D-Plot и далее обычные операции построения графика, построим в окне 2 график параметрической функции.
5. С помощью команды Window Next перейдем в окно 3, т. е. сделаем его активным. Выделим в нем выражение #3. Используя команды Window Designate 3D-Plot, подготовим окно к построению в нем графика трехмерной поверхности. Используя команду Plot и подобрав подходящие масштабы, построим график функции по выражению #3.

В результате этих построений получим рис. 1.16. Он дает наглядное представление о всех трех основных формах графических построений, реализованных в системе Derive.

Derive позволяет строить и графики специального типа. Примером могут служить логарифмические амплитудно-частотные характеристики (ЛАЧХ) и диаграммы Бодэ на комплексной плоскости, широко используемые при анализе линейных электро- и радиотехнических цепей и систем автоматического управления, заданных их операторными передаточными функциями. Следующий пример иллюстрирует это.

Рис. 1.16
Вид построений



Пусть требуется построить ЛАЧХ и диаграмму Бодэ для системы, описываемой операторной передаточной функцией следующего вида:

$$F(p) := \exp(-p) / (p^2 + 0.8 * p + 1)$$

Чтобы построить ЛАЧХ, достаточно задать логарифмические масштабы по осям графика (с основанием логарифма 10). При этом следует ввести выражение:

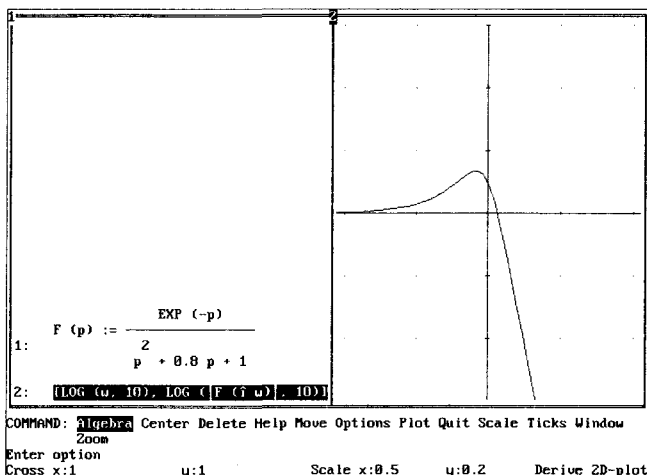
$$[\log(w, 10), \log(\text{abs}(F(\#i * w)), 10)]$$

Заметим, что тут оператор p заменяется на $\#i * w$, где w — круговая частота, и вычисляется логарифм модуля передаточной характеристики.

Для построения диаграммы Бодэ достаточно ввести выражение:

$$[\text{re}(F(\#i * w)), \text{im}(F(\#i * w))]$$

Рис. 1.17
Исходные формулы, ЛАЧХ и диаграмма Бодэ



Разбив экран на три окна, введя эти выражения в большое окно и построив (как описывалось выше) графики функций по двум последним выражениям, получим документ, показанный на рис. 1.17.

Напоминаем, что Derive не позволяет записать этот рисунок в виде графического файла на диск. Используйте для этого отдельные резидентные программы. С помощью соответствующих программ (например, драйвера `prtscr.com`) можно напечатать изображение рисунков на принтере.

Представление результатов вычислений в нескольких окнах делает их более наглядными. Для получения достаточно информативных и красочных многооконных изображений необходимы определенные практические навыки работы с системой. Они легко достигаются, стоит лишь самостоятельно построить графики нескольких функций.

1.22. Команда численных вычислений (approx)

Команда `approx` используется для приближенных вычислений математических выражений. Слово “приближенные” не следует воспринимать в смысле “грубые” или “неточные”. В действительности алгоритм вычислений гаранти-

рует правильность всех цифр результата, за исключением последней в дробных числах.

При использовании команды `арргоX` появляется запрос:

APPROXIMATE expression #n

В ответ на него следует указать номер строки `n`, в которой находится вычисляемое выражение. Ограничимся этим описанием команды `арргоX`, поскольку многочисленные конкретные примеры ее применения были приведены выше.

1.23. Сообщения об ошибках

Как отмечалось, Derive содержит минимум сообщений об ошибках. Ниже указаны основные из них с русским переводом:

Сообщение	Перевод сообщения
Zero Divide	Деление на ноль (результат inf)
Undefined Functuo	Неопределенная функция
Syntax Error	Синтаксическая ошибка
Non-list Object	Не лист с объектами
Nonsymbolic Argument	Не символьный аргумент
Nonnumeric Argument	Не численный аргумент
Noninteger Argument	Не целочисленный аргумент
No Solution found	Не найдено условие
Insufficient Argument	Недостаточный аргумент
Memory Full	Исчерпана память ОЗУ
Disk Full	Диск заполнен
End_Of_File	Конец файла
Console Interrupt	Прерывание с консоли (клавиатуры)

Эти сообщения появляются в нижней части экрана. Если сообщение возникает при вводе математического выражения, то ввод приостанавливается и курсор указывает на возможное место ошибки. Оно будет правильным, если набрано правильное имя функции. В противном случае Derive может воспринять вводимые знаки как имя переменной, а частично как имя определенной функции. Если синтаксически введенное выражение верно, но заданная операция невыполнима (такие ситуации встречаются довольно часто), то система просто повторяет выражение в очередной строке или выводит знак вопроса.

Малочисленность сообщений об ошибках связана с двумя факторами. Во-первых, многие ситуации, которые в других (не символьных) системах ошибочны, в Derive корректны. Например, задание вычислений символьных выражений в общей форме. Во-вторых, практически невозможно указать все ошибки в задании символьных выражений и тем более проанализировать их. Поэтому принятая в Derive реакция (повтор выражения) вполне обоснованна. Следует иметь в виду, что работа с символьными операциями предполагает достаточно высокую математическую культуру у пользователя и его способность критически интерпретировать результаты вычислений.

Память ПК ограничена. Между тем спецификой символьных вычислений нередко является разбухание результатов, как конечных, так и промежуточных. Обычно так происходит при использовании команды `Simplify` (явном

или в составе других команд). Поэтому в системе реализовано динамическое распределение памяти и ее постоянный контроль (в процентах). Сообщение Memory Full свидетельствует об исчерпании памяти, а сообщение Disk Full появляется при попытке записи файлов на диск, который полностью заполнен другими файлами.

Сообщение Memory Full довольно часто появлялось в ранних версиях Derive. Поскольку версия ХМ 3.11 работает с памятью большого объема (до 4 Гбайт), возникновение такого сообщения у современных ПК с реальным объемом ОЗУ в 16–32 Мбайта маловероятно, хотя и не исключено при решении сложных задач.

Глава 2. Язык программирования системы Derive

2.1. Алфавит системы и комментарии

2.1.1. Понятие о входном языке Derive

Derive создавалась как система, не требующая программирования при решении многих математических задач. Однако это не означает, что программирование тут не нужно вообще. Какой бы мощной ни была математическая система, она не может решать любые задачи. Хотя бы потому, что постепенно возникают все новые и новые задачи, о которых создатели системы прежде ничего не знали. Поэтому все серьезные системы имеют свой специфический язык программирования, позволяющий описать решение задач, выходящих за пределы встроенных возможностей системы. Этот язык в отличие от языка реализации muLISP, на котором написана сама система, называется входным языком.

Derive (см. главу 1) позволяет реализовать многие массовые виды вычислений в командном (непосредственном) режиме. Однако все эти виды вычислений можно задать и на входном языке программирования системы, выполняя их в программном режиме работы. Именно так организованы многие сложные вычисления. Программы, реализующие их, можно загружать в систему с использованием команд Load, Merge и Demo.

Итак, входной язык Derive рассчитан прежде всего на работу в режиме прямых вычислений. Поэтому он включает в себя набор математических функций и функций для различных математических преобразований. В основном входной язык ориентирован на применение линейных алгоритмов. Для реализации разветвляющихся алгоритмов язык содержит довольно скромные средства, впрочем, вполне достаточные для решения большинства математических задач при определенном (функциональном) подходе к их решению.

Входной язык Derive относится к классу интерпретирующих. Это означает, что он анализирует каждую введенную инструкцию и тут же ее исполняет.

Интерпретирующий режим работы принят в подавляющем большинстве математических систем, лишь Mathematica 2/3 имеет возможности компиляции своих отдельных программных конструкций и быстрого их исполнения. Интерпретирующий режим роднит Derive с Бейсиком. Но если Бейсик — универсальный язык программирования с весьма посредственными математическими возможностями, то Derive — язык функционального программирования сверхвысокого уровня, проблемно-ориентированный на решение математических задач. Derive содержит множество встроенных функций (например, для вычисления пределов, производных, интегралов и пр.), которые при реализации их на обычных языках программирования потребовали бы многих месяцев напряженной работы программистов.

Программы для системы Derive хранятся в виде программных файлов. Они могут быть двух видов. Файлы с расширением .mth после загрузки требуют запуска. Демонстрационные файлы с расширением .dmo обладают автозапуском после загрузки, каждая их строка выполняется с приостановкой до

нажатия любой клавиши (после чего выполняется следующая строка). Файлы имеют текстовый формат и могут создаваться как внутри самой системы, так и с помощью любого текстового редактора, создающего файлы в формате ASC II. Они могут содержать лишь допустимые для входного языка Derive символы и слова, в совокупности образующие алфавит этого языка.

2.1.2. Алфавит Derive

В алфавит входного языка системы Derive входят латинские буквы от а до z (и от А до Z), арабские цифры от 0 до 9 и все доступные для ввода символы клавиш клавиатуры. В первую очередь это знаки математических операций, греческие буквы (см. раздел 2.4) и спецзнаки:

#e	Alt + E	— основание натурального логарифма	(И);
#i	Alt + I	— мнимая единица (квадратный корень из -1)	(М);
pi	Alt + P	— площадь единичного круга	(y);
Alt-Q	SQRT	— квадратный корень.	

Для отличия от обычных знаков спецзнаки e и i на экране дисплея выделяются символом птички ^ над ними, что говорит об особом статусе этих букв. Буква в скобках указывает на то, как отображается знак при использовании драйвера русификации системы.

2.1.3. Проблемы русификации

К сожалению, при альтернативной кодировке знакогенераторов дисплея и принтера в оригинальной версии Derive вместо знаков e и i отображаются русские буквы И и М, что делает выражения, содержащие экспоненциальные и комплексные функции, мягко скажем, не совсем обычными (к счастью, в текстах, выводимых принтером, этого недостатка нет — мнимая единица выводится как #i, а буква e словом exp). Искажаются и другие знаки, в частности часть греческих букв и изображение интеграла.

Если в ПК загружен драйвер кириллицы, то (как отмечалось) использование большей части греческих букв становится невозможным, так как их образы в знакогенераторах видеоадаптера и принтера заменяются образами букв русского алфавита. Зато эти буквы можно использовать в текстовых комментариях, задаваемых в кавычках.

При подготовке новых программ для системы Derive рекомендуется избегать использования греческих букв. Вместо них можно использовать переменные с созвучными именами, например alpha, beta и т. д. Однако некоторые сочетания букв, например тау, система сама пытается вывести на экран дисплея как греческие буквы.

Знать соответствие между греческими буквами и символами кириллицы полезно при анализе зарубежных программ, запускаемых на компьютерах с русификацией знакогенераторов видеоадаптера. Вообще говоря, ради достоверности отображения всех математических знаков разумно отказаться от применения драйвера русификации дисплея и перейти к комментариям на исконно английском языке.

2.1.4. Текстовые комментарии

Текстовые комментарии могут быть двух видов. В режиме Author можно вводить строки текста, заключенные в кавычки, например:

"Это пример символьного преобразования многочлена"

Такая строка после ввода становится пронумерованной строкой в верхней части экрана дисплея. В самом программном файле номера строк математических выражений и строк с комментариями отсутствуют. Система сама проставляет номера строк при выводе их на экран дисплея. Вообще говоря, указанная строка является особой переменной, чем-то вроде символьных констант в Бейсике.

После знака ; (точка с запятой) можно задавать служебные комментарии — аннотации. Они выводятся по ходу вычислений в нижнюю часть экрана вместо временно исчезающего главного меню. Так можно комментировать ход вычислений по каждой строке, не засоряя верхнюю часть экрана большим числом пояснений. Этот прием использован в демонстрационных программах Derive, и вы можете применить его в своих примерах. Для ввода подобных комментариев годится любой текстовый редактор, поддерживающий формат ASCII.

2.1.5. Константы

Константы — это имеющие имя объекты, значения которых неизменны и автоматически задаются перед началом работы с системой. В алфавит Derive входят три слова-константы:

pi и **π** — число "пи" 3.14159265 (вводится как Alt + P);

deg — константа ($\pi/180$) для перевода углов из градусов в радианы;

inf — положительная машинная бесконечность.

Эти и числовые константы вместе с функциями могут входить в состав арифметических выражений, например:

$$5 * \sin(2 * \pi * x + \text{phi})$$

Здесь 5, 2 и π — константы, \sin — наименование функции с аргументом в круглых скобках, а phi — имя переменной (см. следующий раздел).

Числа относятся к константам. Они могут быть целыми (123, -45 и т. д.), десятичными (123.45, 3.75, -0.005) с разделительной точкой или представленными в экспоненциальном формате с мантиссой и порядком ($-123.456 * 10^5$ или $432.25 * 10^{-12}$). Для представления комплексных чисел используется константа #i, отображаемая дисплеем как i с птичкой сверху. Константа #e (на экране дисплея e с птичкой сверху) является основанием натурального логарифма.

2.2. Переменные и функции пользователя

2.2.1. Переменные и их определение

Переменные в Derive могут быть объявленными и не объявленными. Необъявленные переменные обычно используются в символьных вычислениях, а объявленные — в численных вычислениях. И те и другие широко применяются в вычислениях. Кроме того, возможно задание областей определения объявленных переменных.

Как и во всех математических системах, объявленные переменные — это поименованные области памяти, используемые под хранение численных или

символьных значений. Такие переменные задаются с помощью знака присваивания ($:=$). Выглядит это так:

Имя_переменной := Выражение

Имя переменной может состоять из одной или нескольких букв и цифр, но начинаться должно с буквы. Имя является идентификатором переменной и должно быть уникальным, т. е. не совпадать с именами операторов, функций и констант системы. Выражение может возвращать единственное значение, вектор или матрицу значений. Как отмечалось, при загрузке драйвера кириллицы возможно применение в идентификаторах букв русского алфавита. Не стоит, однако, широко практиковать это, так как из-за различия драйверов (или при их отключении) идентификаторы теряют смысл и выводятся на экран дисплея в виде форменной абракадабры.

Рекомендуется использовать значащие имена переменных, например *Massa* — для массы, *Xcoordinate* — для координаты X, и т. д. Можно составить имя переменной и из нескольких слов, разделяя их знаком `_`, например *Bessel_func*. Независимо от того, какими (прописными или строчными) буквами набрано имя, *Derive* представляет имена переменных, векторов и матриц строчными буквами. Заданные таким образом переменные являются глобальными, т. е. их значения можно изменить и использовать в любом месте программы.

Латинские буквы от *a* до *z* *Derive* резервирует как предопределенные переменные. Это означает, что их можно использовать в символьных выражениях без определения. При этом в выражениях можно не проставлять знак умножения, например $(a\ b\ c)$ в выражении будет означать $(a * b * c)$. Большинство греческих букв являются идентификаторами предопределенных переменных. Но некоторые, например константа π или обозначение гамма-функции Γ , зарезервированы под специальные цели.

Однако описанным способом нельзя действовать с переменными, имена которых состоят из большего числа символов. Например, если ввести выражение $\exp(abc)$, не определив переменную *abc*, то выражение после ввода появится в виде $\text{EXP}(a\ b\ c)$, т. е., по сути, будет $\exp(a * b * c)$. Поэтому переменные с составным именем должны предварительно объявляться (см. раздел 1.7).

В некоторых символьных операциях вводится понятие приоритета переменных. Например, если выражение содержит несколько переменных (*x*, *y*, *z* и т. д.), оно может быть разложено или сгруппировано относительно одной из переменных или всех с определенным приоритетом. По умолчанию задан порядок переменных *x*, *y* и *z*, но этот порядок при желании можно изменить.

2.2.2. Функции пользователя и их декларация

В программировании под функцией понимают некоторую вычислительную операцию над заданными параметрами, возвращающую в ответ на обращение к ней единственный результат. Это может быть число того или иного типа (например, действительное или комплексное) или символьное выражение, полученное в ходе заданных операций. Функции бывают встроенные и задаваемые пользователем.

Несмотря на обилие встроенных в систему функций (экспоненциальных, показательных, тригонометрических, статистических и иных), всегда может потребоваться та или иная новая функция. Поэтому в *Derive* содержится воз-

возможность задания новых функций в виде функций пользователя. Тут используется следующая конструкция:

Имя_функции(Список переменных) := Выражение

Список переменных (параметров функции) является набором имен переменных, разделенных запятыми. При этом переменные, входящие в список, локальны, т. е. область их действия ограничена выражением, определяющим функцию. Например, в функции

CS(x,y) := sin(x) + cos(y)

переменные x и y используются только в рамках этой конструкции. Если задано обращение к функции $CS(1, 2)$, переменная x будет иметь значение 1, а y значение 2 только в выражении $\sin(x) + \cos(y)$, представляющем собой тело функции. В самой же программе эти переменные могут обозначать все, что угодно, например координаты точки, и иметь иные произвольные значения. Имя функции Derive представляет прописными буквами.

Декларированные переменные Derive являются глобальными переменными, т. е. их значения могут меняться в любом месте программы (документа) и внутри функций пользователя, если они используют эти переменные. Из приведенного примера видно, что глобальные и локальные переменные могут иметь совпадающие имена. Это облегчает программирование. Тем не менее правилом хорошего тона при программировании является назначение различных имен для локальных и глобальных переменных.

Функции пользователя, как и встроенные функции, возвращают в ответ на обращение к ним единственное числовое значение, вектор или матрицу. Поэтому их можно включать в состав арифметических выражений, например:

P := 2 * CS(pi / 2, 2 * pi) - 1

Обратите внимание на то, что параметры функции могут задаваться арифметическими выражениями.

2.2.3. Примеры применения объявленных переменных и функций

Следующий пример иллюстрирует задание двух переменных a и b и двух функций пользователя.

1: "Задание переменных и функций пользователя"

2: $a := 2$

3: $b := 5$

4: $LINAPPR(a, b, x) := a + b \cdot x$

5: $LINAPPR(a, b, 0.5)$

6: 4.5

7: $LINAPPR(a, b, 1)$

8: 7

9: FUNC (x, y) := a x + b y

10: FUNC (1, 2)

11: 12

12: LINAPPR (1, 1, 1)

13: 2

14: a

15: 2

16: b

17: 5

Действия здесь достаточно очевидны. В строках 12–17 приведен пример, наглядно проясняющий различные статусы переменных a и b . Вначале (в строках 2 и 3) они заданы как глобальные переменные, а затем используются как локальные переменные в функции `LINAPPR`.

Во всех приведенных здесь примерах параметрами функций были числовые значения, имена переменных или арифметические выражения. Важно, чтобы они состояли из уже заданных определений. А можно ли в качестве параметров использовать символьные выражения, описывающие произвольную и еще не определенную функцию? Вообще говоря, нет. Однако в `Derive` существует особое соглашение на этот счет. Можно произвольную функцию в списке параметров заданной функции обозначить любой буквой и передавать ее символьное значение в тело функции, содержащее, в свою очередь, только заведомо определенные функции. Этот особый и принципиально важный случай заслуживает особого рассмотрения, выполненного в разделе 2.9.

2.3. Арифметические и логические операторы и функции

2.3.1. Форматы чисел

При арифметических операциях `Derive` может работать как с действительными, так и комплексными числами. О действительных числах-константах уже говорилось. Обратимся теперь к комплексным числам.

Пусть z и w означают два комплексных числа или две переменные, имеющие комплексное значение:

$$\begin{aligned} z &= \operatorname{Re}(z) + i * \operatorname{Im}(z), \\ w &= \operatorname{Re}(w) + i * \operatorname{Im}(w), \end{aligned}$$

где Re и Im — действительная и мнимая части z и w , а i — мнимая единица (квадратный корень из -1). В такой форме `Derive` оперирует с комплексными числами.

2.3.2. Арифметические операторы

Для выполнения арифметических операций используются операторы в виде обычных математических знаков:

- $-z$ — смена знака у z ;
- $z + w$ — сложение z и w ;
- $z - w$ — разность между z и w ;
- $z * w$ — умножение z на w ;
- $z w$ — умножение z на w ;
- z / w — деление z на w ;
- $z ^ w$ — возведение z в степень w ;
- $z \%$ — вычисление $z/100$.

Ввиду очевидности этих операций они не нуждаются в особых комментариях. Следует, однако, отметить, что в сложных выражениях операции выполняются с общепринятым приоритетом: вначале выполняются операции возведения в степень, умножения и деления, а затем сложения и вычитания. Операция вычисления процентов обладает более высоким приоритетом, но еще выше логические операции и вычисление функций. При задании комплексных чисел мнимая единица вводится как $\#i$. Приведем примеры на арифметические операции с комплексными числами.

1: "Примеры операций с комплексными числами"

2: $z := a + \#i b$

3: $w := c + \#i d$

4: $z + w$

5: $a + c + \#i (b + d)$

6: $z w$

7: $a c - b d + \#i (a d + b c)$

8:
$$\frac{z}{w}$$

9:
$$\frac{a c}{c^2 + d^2} + \frac{b d}{c^2 + d^2} + \#i \left(\frac{b c}{c^2 + d^2} - \frac{a d}{c^2 + d^2} \right)$$

10: RE
$$\left(\frac{a c}{c^2 + d^2} + \frac{b d}{c^2 + d^2} + \#i \left(\frac{b c}{c^2 + d^2} - \frac{a d}{c^2 + d^2} \right) \right)$$

11:
$$\frac{a c}{c^2 + d^2} + \frac{b d}{c^2 + d^2}$$

12:
$$\text{IM} \left(\frac{ac}{c+d} + \frac{bd}{c+d} + \#i \left(\frac{bc}{c+d} - \frac{ad}{c+d} \right) \right)$$

13:
$$\frac{bc}{c+d} - \frac{ad}{c+d}$$

14: PHASE (3 + 2 #i)

15:
$$\begin{array}{r} 31661 \\ \text{-----} \\ 53845 \end{array}$$

16: CONJ (3 + 2 #i)

17: 3 - 2 #i

18:
$$\frac{(5 - \#i 3) (3 + \#i 2)}{(5 + \#i 3) (2 - \#i 4)} + (0.5 + \#i 1)$$

19:
$$\frac{197}{170} + \frac{249}{170} \#i$$

2.3.3. Логические операторы

Пусть u и v — некоторые выражения или просто переменные, имеющие определенные значения. Для выражения отношений их равенства или неравенства используются следующие операторы:

- $u = v$ — равенство выражений;
- $u \neq v$ — неравенство выражений;
- $u < v$ — выражение u меньше v ;
- $u > v$ — выражение u больше v ;
- $u \leq v$ — выражение u меньше или равно v ;
- $u \geq v$ — выражение u больше или равно v .

Указанные выше конструкции дают результат TRUE (ИСТИННО) или FALSE (ЛОЖНО), который трактуется как логический. Derive включает три логических оператора, кстати, не отмеченных в системе оперативной справки. Если Y — условие, то эти операторы используются в виде:

- NOT Y** — логическое отрицание;
- Y1 AND Y2** — логическое сложение;
- Y1 OR Y2** — логическое умножение.

Действие этих операторов легко понять, считая, что TRUE дает логическую единицу 1, а FALSE — логический ноль 0. Тогда NOT дает 1, если $Y = 0$, и 0, если $Y = 1$. Таблица истинности операторов AND и OR такова:

Y1	Y2	AND	OR
0	0	0	0
1	0	1	0
0	1	1	0
1	1	1	1

0 - FALSE (ЛОЖНО)
1 - TRUE (ИСТИННО)

Здесь учитывается, что в логическом смысле $0 + 1 = 1$, $1 + 1 = 1$, $0 * 1 = 0$ и $1 * 1 = 1$. Не следует путать логические значения с численными значениями.

2.3.4. Функция IF для создания условных выражений

При программировании ряда задач нужна функция, действие которой зависит от определенных условий. В Derive такой функцией является функция IF (“если”). Работа ее довольно проста: если условие выполняется, она возвращает одно значение, а если условие не выполняется — другое.

Синтаксис задания условной функции IF в Derive следующий:

IF (Y, u, v)

Если условие Y дает TRUE, эта функция возвращает значение выражения u, а если условие дает FALSE — возвращается значение выражения v.

В общем случае можно использовать следующие конструкции условной функции:

- IF (r, t, f)** — если отношение r истинно, выдается выражение t; если оно ложно, выдается выражение f;
- IF (r, t, f, u)** — если отношение r истинно, выдается выражение t; если оно ложно, выдается выражение f; если истинность неизвестна, выдается u;
- IF (r AND s, t, f)** — если отношение r ложно, выдается выражение f; в противном случае выдается IF(s, t, f);
- IF (r OR s, t, f)** — если отношение r истинно, выдается выражение t; в противном случае выдается IF(s, t, f);
- IF (NOT r, t, f)** — если отношение r ложно, выдается выражение t; если оно истинно, выдается выражение f.

Ниже дан пример на действие логических операторов и функции IF.

- ```

1: "Логические операторы и функции"
2: IF (1 = 1, 1, 0)
3: 1
4: IF (1 = 0, 1, 0)
5: 0
6: IF (NOT 1 = 1, 1, 0)
7: 0
8: IF (1 = 1 AND 1 = 1, 1, 0)
9: 1
10: IF (1 = 1 AND 1 = 0, 1, 0)
11: 0
12: IF (1 = 1 OR 1 = 0, 1, 0)

```

```

13: 1
14: IF (1 = 0 OR 0 = 1, 1, 0)
15: 0

```

Используя функцию IF, можно легко создавать разрывные функции или функции, описывающие специальные сигналы. Пример этого приведен ниже:

```

1: "Задание функции - выпрямление синусоиды"
2: F (x) := IF (SIN (x) > 0, SIN (x), 0)
3: F (1)
4: 6842

 8131
5: F (-1)
6: 0
7: SIN (-1)
8: 6842

 8131

```

Контрольные примеры, приведенные в строках от 3 до 8, показывают, что здесь отсутствует отрицательная полуволна синусоидального сигнала. В этом можно убедиться еще нагляднее, построив график функции F(x).

## 2.4. Математические функции

Система Derive содержит довольно много встроенных элементарных математических функций. Их несколько больше, чем нужно для представления всего набора элементарных функций. Например, функция  $\tan(z)$  излишняя, так как выражается в виде  $\sin(x)/\cos(x)$ . Тем не менее ради удобства работы состав встроенных элементарных функций сделан достаточно полным.

Признаком функции является возврат значения, численного или символьного. Поэтому функции могут использоваться совместно с операторами для записи математических выражений, вычисляемых Derive. Ниже встроенные в ядро Derive математические функции квалифицируются по группам. Отображаются функции, в отличие от переменных, прописными буквами, например SIN(x), но набираться могут буквами любых размеров.

### 2.4.1. Числовые функции

Числовые функции (x и y имеют действительные значения):

|                        |                                                                   |
|------------------------|-------------------------------------------------------------------|
| <b>ABS (x)</b>         | — возвращает абсолютное значение x;                               |
| <b>SIGN (x)</b>        | — возвращает 1 при $x > 0$ ; 0 при $x = 0$ и $-1$ при $x < 0$ ;   |
| <b>MAX (x, y, ...)</b> | — возвращает максимальное значение из списка значений аргументов; |

- MIN (x, y, ...)** — возвращает минимальное значение из списка значений аргументов;
- STEP (x)** — возвращает 1 при  $x > 0$  и 0 при  $x < 0$ ;
- CHI (a, x, b)** — возвращает 1 при  $a < x < b$  и 0 при  $x < a$  or  $x > b$ ;
- NEXT\_PRIME (n)** — возвращает следующее простое число; превосходящее  $n$ .

В отношении числовых функций при их упрощении действуют следующие соотношения:

$$\text{MAX (x, y)} \rightarrow \frac{|x - y|}{2} + \frac{x + y}{2},$$

$$\text{MIN (x, y)} \rightarrow \frac{x + y}{2} - \frac{|x - y|}{2},$$

$$\text{STEP (x)} \rightarrow \frac{\text{SIGN}(x)}{2} + \frac{1}{2},$$

$$\text{CHI (a, x, b)} \rightarrow \frac{\text{SIGN}(x - a)}{2} - \frac{\text{SIGN}(x - b)}{2}.$$

## 2.4.2. Кусочно-непрерывные функции

Кусочно-непрерывные функции — это функции, значения которых могут резко различаться в точках разрывов, но в промежутках между ними они изменяются плавно и являются непрерывными. В Derive определены следующие кусочно-непрерывные функции:

- FLOOR (m, n)** — возвращает наибольшее целое, меньшее или равное  $m/n$ ;
- FLOOR (m)** — возвращает целую часть  $m$ ;
- MOD (m, n)** — возвращает  $m$  по модулю  $n$  (неотрицательный остаток от  $m/n$ );
- MOD (m)** — возвращает дробную часть  $m$ ;
- MODS (m, n)** — возвращает симметричное  $m$  по модулю  $n$ , которое оказывается в полуоткрытом интервале  $[-n/2, n/2)$ , причем по умолчанию  $n = 1$ ;
- GCD (m, n, ...)** — возвращает наибольший общий делитель  $m, n, \dots$ ;
- LCM (m, n, ...)** — возвращает наименьшее общее кратное  $m, n, \dots$ .

Если  $m$  и  $n$  не числа, то функции **MOD (m, n)** и **MODS(m, n)** упрощаются к следующим выражениям:

$$\text{MOD (m, n)} \rightarrow m - n \text{ FLOOR (m, n)}$$

$$\text{MODS (m, n)} \rightarrow m - n \text{ FLOOR}(m + n/2, n)$$

## 2.4.3. Функция генерации случайных чисел

Некоторые виды анализа и моделирования различных явлений и систем требуют применения случайных чисел. В Derive содержится функция генерации случайных чисел с равномерным распределением

**RANDOM (n),**

которая в зависимости от значения аргумента  $n$  выполняет следующие действия:

$n > 1$  — возвращает случайное целое в интервале  $[0, n]$ ;

$n = 1$  — возвращает случайное число в интервале  $[0, 1]$ ;

$n < 1$  — устанавливает счетчик генератора случайных чисел в  $-n$ ;

$n = 0$  — устанавливает счетчик генератора случайных чисел по текущему времени (прошедшему с момента загрузки Derive) в сотых долях секунды.

Примеры использования этих функций представлены ниже.

1: "Примеры операций с числовыми функциями"

2:  $x := -10$

3:  $|x|$  SIGN ( $x$ )

4: -10

5: MAX (1, 2, 3, 4, 5)

6: 5

7: MIN (1, 2, 3, 4, 5)  $x$

8: -10

9:  $Y(x) := 5 \times$  STEP ( $x$ )

10:  $Y(1)$

11: 5

12:  $Y(-1)$

13: 0

14:  $F(x) := 5 \times$  CHI (1,  $x$ , 3)

15:  $F(0.5)$

16: 0

17:  $F(2)$

18: 10

19:  $F(5)$

20: 0

21: NEXT\_PRIME (100)

22: 101

23: NEXT\_PRIME (101)

24: 103

25. FLOOR (4.75)

26: 4

27: GCD (6, 15)

28: 3

29: LCM ([6, 15, 14])

30: 210

#### 2.4.4. Степенные (экспоненциальные) функции

Степенные (экспоненциальные) функции относятся к классу алгебраических функций. В Derive определены основные из них:

**SQRT (z)** — квадратный корень из  $z$  (на дисплее отображается как  $\sqrt{z}$ );

**#e** — основание натурального логарифма;

**EXP (z)** — число #e в степени  $z$ .

При комплексном аргументе  $z$  фаза квадратного корня из  $z$  может находиться в пределах от  $-PI/2$  до  $+PI/2$  радиан. Преобразование

$$\#e^z \#e^w \leftrightarrow \#e^{z+w} \quad (2.1)$$

справедливо, если  $z$  и  $w$  действительные или комплексные числа. Преобразование

$$(\#e)^{zk} \leftrightarrow \#e^{kz} \quad (2.2)$$

справедливо, если  $k$  — целое или если  $k$  — дробное, а  $z$  — действительное или комплексное выражение с мнимой частью, значение которой лежит в пределах от  $-PI$  до  $+PI$ .

Для указания направления этих преобразований используется команда **Manage Exponential** с направлениями:

**Auto** — (2.1) слева направо, а (2.2) справа налево;

**Collect** — оба преобразования слева направо;

**Expand** — оба преобразования справа налево.

Следует отметить, что результаты символьных операций во многом зависят от направления преобразования. Поэтому рекомендуем практически поупражняться в различных преобразованиях, используя разные их направления.

#### 2.4.5. Логарифмические функции

Логарифмические функции также относятся к классу алгебраических функций. Они возвращают:

**LN (z)** — главное значение натурального логарифма  $z$ ;

**LOG (z, w)** — логарифм  $z$  по основанию  $w$ .

С помощью команды **Manage Logarithm** возможны такие преобразования:

$$LN(x) + LN(z) \leftrightarrow LN(xz) \quad (2.3)$$

$$k LN(x) \leftrightarrow LN(x^k) \quad (2.4)$$

Направления преобразования следующие:

**Auto** — (2.3) слева направо, а (2.4) справа налево;

**Collect** — оба преобразования слева направо;

**Expand** — оба преобразования справа налево.

Здесь также надо помнить, что результат преобразований (а нередко и сама возможность преобразования) зависят от установленного направления преобразования. Это надо учитывать при практической работе с Derive.

## 2.4.6. Тригонометрические функции

Тригонометрические функции — еще один важный класс функций, с которыми работает система Derive. Эти функции возвращают:

**SIN (z)** — синус угла z;

**COS (z)** — косинус угла z;

**TAN (z)** — тангенс угла z;

**COT (z)** — котангенс угла z;

**SEC (z)** — секанс угла z;

**CSC (z)** — косеканс угла z.

Угол z обычно задается в радианах. Но его можно задавать и в градусах, умножая на преобразующую константу deg, например  $\text{SIN}(z \text{ deg})$ , где z — угол, заданный в градусах.

При преобразованиях тригонометрических функций действуют соотношения:

$$\text{TAN}(z) = \text{SIN}(z)/\text{COS}(z) \quad \text{COT}(z) = \text{COS}(z)/\text{SIN}(z)$$

$$\text{SEC}(z) = 1/\text{COS}(z) \quad \text{CSC}(z) = 1/\text{SIN}(z)$$

При упрощении выражений с тригонометрическими функциями используется команда `Manage Trigonometry`. Она вызывает меню:

`MANAGE TRIG: Direction: Auto Collect Expand`  
(направление преобразования)  
`Toward: Auto Sines Cosines`  
(преобразование степеней SIN в COS и наоборот)

Например, для преобразования сумм углов в кратные углы надо выбрать направление `Expand` (расширить):

$$\text{SIN}(z+w) \rightarrow \text{SIN}(z) \text{COS}(w) + \text{COS}(z) \text{SIN}(w)$$

Аналогично

$$\text{SIN}(m z) \rightarrow 2 \text{COS}((m-1) z) \text{SIN}(z) + \text{SIN}((m-2) z)$$

где m — положительное целое, большее 1.

Используйте `Toward Sines` для преобразований вида

$$\text{COS}(z)^2 \rightarrow 1 - \text{SIN}(z)^2$$

и `Toward Cosines` для преобразований вида

$$\text{SIN}(z)^2 \rightarrow 1 - \text{COS}(z)^2$$

Программы, использующие такое указание, могут работать медленно. Однако возможны выражения, где такое явное указание может привести к результату, который не удастся получить с указанием `Toward Auto`.

### 2.4.7. Обратные тригонометрические функции

Обратные тригонометрические функции возвращают значение угла, выраженного в радианах. В Derive используются следующие обратные тригонометрические функции:

- ATAN (z)** — арктангенс  $z$ ;  
**ATAN (y, x)** — арктангенс радиус-вектора точки  $(x, y)$ , причем угол измеряется от положительной полуоси  $x$ ;  
**ACOT (z)** — арккотангенс  $z$ ;  
**ACOT (x, y)** — арккотангенс радиус-вектора точки  $(x, y)$ ;  
**ASIN (z)** — арксинус  $z$ ;  
**ACOS (z)** — арккосинус  $z$ ;  
**ASEC (z)** — арксеканс  $z$ ;  
**ACSC (z)** — арккосеканс  $z$ .

Результат этих функций — угол в радианах, приведенный в диапазоне от  $-\pi$  до  $+\pi$ . Исключением является угол, возвращаемый функцией **ATAN(y, x)**; он лежит в области значений от  $-\pi/2$  до  $+\pi/2$ . В этом же диапазоне углов лежат значения **ASIN(x)**, если  $-1 \leq x \leq +1$ . При тех же значениях  $x$  значение **ACOS(x)** лежит в пределах от  $0$  до  $\pi$ . При преобразованиях такого класса функций действуют следующие соотношения:

$$\begin{aligned} \text{ACOT}(z) &= \pi/2 - \text{ATAN}(z) & \text{ACOT}(x, y) &= \text{ATAN}(y, x) \\ \text{ATAN}(y, x) &= \text{ATAN}(y/x) & \text{ACOS}(z) &= \pi/2 - \text{ASIN}(z) \\ \text{ASEC}(z) &= \text{ACOS}(1/z) & \text{ACSC}(z) &= \text{ASIN}(1/z) \end{aligned}$$

Если желателен возврат угла этими функциями в градусах, поделите возвращаемое значение на константу `deg`.

### 2.4.8. Гиперболические функции

Derive поддерживает работу со следующими гиперболическими функциями:

- SINH (z)** — гиперболический синус  $z$ ;  
**COSH (z)** — гиперболический косинус  $z$ ;  
**TANH (z)** — гиперболический тангенс  $z$ ;  
**COTH (z)** — гиперболический котангенс  $z$ ;  
**SECH (z)** — гиперболический секанс  $z$ ;  
**CSCH (z)** — гиперболический косеканс  $z$ .

Гиперболические функции упрощаются в выражения, содержащие экспоненциальные функции, например:

$$\text{SINH}(z) = \frac{e^z}{2} - \frac{e^{-z}}{2} \quad \text{и т. д.}$$

Другие подобные определения гиперболических функций общеизвестны, и мы не будем их приводить.

### 2.4.9. Обратные гиперболические функции

Derive поддерживает следующие обратные гиперболические функции:

- ASINH (z)** — обратный гиперболический синус z;  
**ACOSH (z)** — обратный гиперболический косинус z;  
**ATANH (z)** — обратный гиперболический тангенс z;  
**ACOTH (z)** — обратный гиперболический котангенс z;  
**ASECH (z)** — обратный гиперболический секанс z;  
**ACSCH (z)** — обратный гиперболический косеканс z.

Обратные гиперболические функции упрощаются к их логарифмическим представлениям, например:

$$\text{ATANH}(z) = \frac{\text{LN}(z+1)}{2} - \frac{\text{LN}(1-z)}{2} \quad \text{и т. д.}$$

Хорошо известны подобные соотношения и для других обратных гиперболических функций.

### 2.4.10. Примеры операций с элементарными функциями

Ниже представлен ряд примеров символьных и числовых вычислений с применением элементарных функций.

1: "Примеры операций с элементарными функциями"

2:  $\text{TANH}(x)$

$$3: \frac{e^{2x} - 1}{e^{2x} + 1}$$

4:  $\text{SIN}(a + i b)$

$$5: \frac{b \text{ SIN}(a)}{2} + \frac{-b \text{ SIN}(a)}{2} + i \left( \frac{b \text{ COS}(a)}{2} - \frac{-b \text{ COS}(a)}{2} \right)$$

6:  $\text{LOG}(1000, 10)$

7: 3

8:  $\text{LN}(a + i b)$

$$9: \frac{\text{LN}(a^2 + b^2)}{2} - i \left( \frac{\text{PI SIGN}(a b)}{2} - \frac{\text{PI SIGN}(b)}{2} - \text{ATAN} \left( \frac{b}{a} \right) \right)$$

10:  $10(1 - \text{EXP}(-2))$

$$11: \frac{14829}{1715}$$



Обратите внимание на значки ~ в конце каждой длинной строки и в начале следующей за ней строки. Эти значки обозначают перенос выражения в длинной строке на следующую строку. Они вставляются системой при использовании команд Transfer Print File, применяемых для вывода документа в текстовый файл, который затем можно распечатать на принтере. На экране дисплея такого переноса нет, что, кстати, неудобно при просмотре длинных выражений или чисел.

### 2.4.11. Функции комплексного аргумента

Функции комплексного аргумента:

- #i** — мнимая единица (Alt + I);
- ABS (z)** — модуль z;
- SIGN (z)** — упрощается к точке на единичной окружности с тем же углом, что у z (при  $z = 0$  — угол любой и упрощения не происходит);
- RE (z)** — действительная часть z;
- IM (z)** — мнимая часть z;
- CONJ (z)** — комплексно-сопряженное с z число;
- PHASE (z)** — фаза — угол радиус-вектора z, определенный в диапазоне значений от  $-\pi$  до  $+\pi$ .

Следующий пример иллюстрирует применение этих функций.

1: "Функции комплексных чисел"

2:  $z := 3 + 2 \#i$

3: SIN (z)

4: 0.530922 - 3.59056 #i

5: RE (SIN (z))

6: 0.530922

7: IM (SIN (z))

8: -3.59056

9: SIGN (z)

10: 0.832050 + 0.554700 #i

11: |z|

12: 3.60555

13:  $\frac{z}{|z|}$

14: 0.832050 + 0.554700 #i

15: PHASE (z)

16: 0.588002

17: CONJ (z)

18: 3 - 2 #i

Помимо элементарных функций Derive включает несколько наиболее распространенных встроенных специальных математических функций. Это вычисление факториала, гамма-функции и функций комбинаторики:

## 2.4.12. Факториальные и комбинаторные функции

Факториальные и комбинаторные функции:

$z!$  — факториал  $z$ ;

**GAMMA** ( $z$ ) или  $\Gamma(z)$  — гамма-функция;

**PERM** ( $n, m$ ) — число размещений из  $n$  элементов по  $m$ ;

**COMB** ( $n, m$ ) — число сочетаний из  $n$  элементов по  $m$ .

Особо следует оговорить использование факториала. По определению, факториал для целого числа  $z$  определяется как

$$z! = 1 \quad \text{при } z = 0 \text{ или } 1$$

$$z! = 1 * 2 * 3 * \dots * z \quad \text{при } z > 1$$

Так Derive вычисляет факториал при действительных целых  $z$ . Однако в общем случае факториал определяется через гамма-функцию:

$$z! = \Gamma(z + 1)$$

Поэтому Derive может вычислять  $z!$  по этому выражению и для любых  $z$ .

Функции **PERM** и **COMB** возвращают следующие значения:

$$\text{PERM} = \frac{n!}{(n-m)!} \quad \text{COMB} = \frac{n!}{(n-m)! * m!}$$

Вычисление указанных выше функций иллюстрируется следующими примерами.

1: "Вычисление факториала"

2: 5!

3: 120

4: 0.5!

5: 0.886226

6: - 2.2!

7: -2.42396

8: "Вычисление гамма-функции"

9: GAMMA (6)

10: 120

- 11: GAMMA (1.5)
- 12: 0.886226
- 13: GAMMA (-3.2)
- 14: 0.689056
- 15: "Вычисление числа размещений"
- 16: PERM (10, 5)
- 17: 30240
- 18: "Вычисление числа сочетаний"
- 19: COMB (10, 5)
- 20: 252

Большое количество специальных математических функций определено во внешних расширениях системы с помощью файлов ее библиотек (глава 3). Нередко эти функции являются решениями дифференциальных уравнений специального вида, значениями интегралов или определениями специальных математических функций, дополняющих встроенные функции.

## 2.5. Функции математического анализа

Значительное количество функций в системе Derive призвано решать типовые задачи математического анализа [31, 32]. В основном это функции, реализующие типовые задачи высшей математики: определение пределов функций, вычисление производных и интегралов, разложение функций в ряд Тейлора и пр. Надо полагать, что читатель, взявшийся за вычисление функций математического анализа, имеет о них достаточное представление.

Ниже представлены форматы использования этих встроенных функций. Тут используются следующие обозначения:

- u** — математическое выражение, описывающее функцию одной или многих переменных;
- x** — текущая переменная (например, та, по которой идет дифференцирование или интегрирование);
- n** — порядок производной или первообразной;
- a, b** — значения  $x$  или пределы интегрирования.

Встроенные в ядро Derive функции математического анализа можно сгруппировать по их назначению.

### 2.5.1. Функции для вычисления пределов

Функции для вычисления пределов:

- LIM (u, x, a)** — предел функции  $u$  по переменной  $x$  в точке  $a$ ;
- LIM (u, x, a, 1)** — предел функции  $u$  по переменной  $x$  выше точки  $a$ ;

**LIM (u, x, a, -1)** — предел функции u по переменной x ниже точки a.

Если задать

$$\text{LIM } (((x + h)^2 - x^2) / ((x + h) - x), h, 0)$$

на экране эта функция предстанет как

$$\lim \frac{(x + h)^2 - x^2}{x + h - x}$$

и упрощается к  $2x$  (производная от  $x^2$ ).

Если функция u разрывная, пределы сверху и снизу могут различаться. Для указания этого вводится четвертый аргумент — в общем случае положительное число для вычисления предела сверху и отрицательное для вычисления предела снизу (удобно вводить соответственно 1 или -1). Бесконечность указывается как inf, например:

$$\text{LIM } (a x / (x + 1), x, \text{inf})$$

изображается как

$$\lim_{x \rightarrow \infty} \frac{a x}{x + 1}$$

и упрощается к a.

Вычисление предела часто бывает более предпочтительным, чем подстановка значений переменных с помощью команды Manage Substitute. Например, для функции  $x/\text{SIN}(x)$  подстановка  $x = 0$  дает неопределенность вида  $0/0$  и упрощается к сообщению ?, тогда как

$$\lim_{x \rightarrow 0} \frac{x}{\text{SIN}(x)}$$

упрощается к 1.

## 2.5.2. Функции дифференцирования

Для дифференцирования используются функции, возвращающие:

**DIF (u, x)** — первую производную функции u по x;

**DIF (u, x, n)** — n-ю частную производную функции u по x.

Например, введя

$$\text{DIF } (\text{SIN}(a x^2), x)$$

получим на экране выражение

$$\frac{d}{dx} \text{SIN } (a x^2)$$

которое упрощается к

$$2 a x \text{COS } (a x^2)$$

Возможно последовательное применение функции DIF для вычисления смешанных производных.

### 2.5.3. Функция разложения в ряд Тейлора

Для вычисления разложения функции в ряд Тейлора используется функция: **TAYLOR** ( $u, x, a, n$ ) — разложение функции  $u$  в  $n$  членов ряда Тейлора при  $x = a$ .

Итог ее работы — степенной многочлен со степенями, убывающими от  $n$  до 0.

### 2.5.4. Функции интегрирования

Ряд функций осуществляют операции интегрирования:

**INT** ( $u, x$ ) — вычисление первообразной интеграла с подынтегральной функцией  $u$  по  $x$ ;

**DIF** ( $u, x, -n$ ) — вычисление первообразной  $n$ -го порядка;

**INT** ( $u, x, a, b$ ) — вычисление определенного интеграла с пределами изменения  $x$  от  $a$  до  $b$ .

Обратите внимание, что функция DIF при записи значения  $n$  со знаком минус возвращает первообразную, а не производную.

### 2.5.5. Функции суммирования и произведения членов рядов

Следующие функции используются для вычисления сумм и произведений:

**SUM** ( $u, n$ ) — сумма ряда функции  $u$  по  $n$ ;

**SUM** ( $u, n, k, m$ ) — сумма ряда функции  $u$  для  $n$  от  $k$  до  $m$ ;

**PRODUCT** ( $u, n$ ) — произведение ряда функции  $u$  по  $n$ ;

**PRODUCT** ( $u, n, k, m$ ) — произведение ряда функции  $u$  для  $n$  от  $k$  до  $m$ .

Примеры применения этих функций были подробно описаны в разделе 1.6. Нет смысла повторять их. Учтите, что возможно последовательное их применение, например для вычисления кратных интегралов. Большое число новых примеров применения этих функций можно найти в главе 4.

### 2.5.6. Функции решения уравнений и неравенств и систем с ними

Четыре функции Derive служат для решения нелинейных уравнений или систем линейных уравнений. В перечисленных ниже функциях  $u$  и  $v$  — арифметические выражения,  $a$  и  $b$  — пределы изменения текущей переменной,  $x$  — корни уравнения (либо  $u = 0$ , либо  $u = v$ ).

Функции для решения уравнений:

**SOLVE** ( $u, x$ ) — вычисление  $x$ , при котором  $u = 0$ ;

**SOLVE** ( $u = v, x$ ) — вычисление  $x$ , при котором  $u = v$ ;

**SOLVE** ( $u = v, x, a, b$ ) — вычисление  $x$  на отрезке  $[a, b]$ , при котором  $u = v$ ;

**SOLVE** ( $u < v, x$ ) — вычисление  $x$  для неравенства  $u < v$ ;

**SOLVE** ( $[u1 = v1, u2 = v2, \dots], [x1, x2, \dots]$ ) — вычисление корней  $x1, x2, \dots$  для системы линейных уравнений.

Ниже представлен пример, показывающий возможности функции SOLVE.

- 1: "Решение уравнений - функция SOLVE"
- 2: SOLVE (x<sup>2</sup> - 3, x)
- 3: 
$$\left| x = \frac{11274}{6509} \right|$$
- 4: SOLVE (x<sup>2</sup> = 3, x)
- 5: 
$$\left| x = \frac{11274}{6509} \right|$$
- 6: SOLVE (SIN (x) = 0, x, 3, 4)
- 7: [x = 3.14160]
- 8: "Решение системы линейных уравнений"
- 9: SOLVE ([3 a+2 b+c=4, a+b-c=1, a-2 b+c=3], [a, b, c])
- 10: [ a = 1.7 b = -0.6 c = 0.1 ]

Первые две задачи этого примера в особых комментариях не нуждаются. Третья задача — строки 6 и 7 — иллюстрирует выбор интервала решения, что важно, когда решение может иметь более одного корня. Именно такое решение и имеет заданное уравнение  $\sin(x) = 0$ . Выбрано очевидное решение  $3 < \pi < 4$ . Наконец, в строках 8–10 показано применение функции SOLVE для решения системы из трех линейных уравнений:

$$\begin{aligned} 3 * a + 2 * b + c &= 4 \\ a + b + c &= 1 \\ a - 2 * b + c &= 3 \end{aligned}$$

с неизвестными a, b и c. Систему линейных уравнений можно решать и с применением матричных операций, но для простых систем описанный выше способ наиболее прост.

### 2.5.7. Функция выполнения итераций (ITERATES)

В практике программирования широко используются структуры, именуемые циклами. Различают циклы с заданным числом повторения операций внутри их и с числом операций, зависящим от выполнения некоторого условия. Оно обычно изменяется в процессе исполнения цикла.

Однако детальный анализ областей применения циклов показывает, что они нужны не сами по себе, а для реализации определенных классов задач, например для суммирования и перемножения членов ряда, создания последовательностей данных и т. д. В Derive эти возможности уже реализованы в функциях SUM и PRODUCT для рядов и VECTOR — для создания последователь-

ностей данных. Остается необходимость в создании итерационных алгоритмов.

Такие алгоритмы в Derive реализует функция ITERATES:

**ITERATES (u, x, x0, n)** — выполнение  $n$  итераций при начальном значении  $x = x_0$  с возвратом результатов каждой итерации;

**ITERATE (u, x, x0, n)** — аналогична ITERATES, но возвращает результат последней итерации.

Функция ITERATES позволяет организовать итерационные циклы вычисления функции  $u(x) = x$ . Если указан параметр  $n$  (число итераций), то функция ITERATES проводит  $n$  итераций. Вначале она подставляет в  $(u(x) - x)$  начальное значение  $x = x_0$  и выполняет первую итерацию, используя простой метод итераций. Затем полученное значение  $x = x_1$  вновь подставляется в  $(u(x) - x)$ , выполняется вторая итерация и т. д. Таким образом, получается вектор значений  $x (x_0, x_1, \dots)$ .

Для уравнений

$$u(x) - x = 0$$

при решении которых метод простых итераций сходится при достаточно большом  $n$ , можно получить ряд значений  $x$ , сходящихся к решению.

Если параметр  $n$  не указан, итерации проводятся до тех пор, пока значения  $x$  на двух последних итерациях не совпадут с заданной системной погрешностью. Она может быть меньше погрешности, обусловленной усечением числа цифр результата, поэтому результат может показать больше двух повторяющихся значений  $x$ .

## 2.5.8. Примеры реализации итерационных вычислений

Действие функций ITERATES и ITERATE поясняют следующие примеры.

1: "Действие функций ITERATES и ITERATE"

2: ITERATES (SIN (x) + 0.25, x, 1.2, 5)

3: "Используем команду approx"

4: [1.2, 1.18203, 1.17538, 1.17283, 1.17185, 1.17147]

5: ITERATES (SIN (x) + 0.25, x, 1.1714)

6: [1.1714, 1.17129, 1.17125, 1.17123, 1.17123, 1.17123, ~  
~1.171123, 1.17122, 1.17122, 1.17122, 1.17122]

7: ITERATE (SIN (x) + 0.25, x, 1.2, 10)

8: 1.17123

9: ITERATE (SIN (x) + 0.25, x, 1.2)

10: 1.17122

В этих примерах решается нелинейное уравнение

$$\text{SIN}(x) + 0.25 = x$$

Функции ITERATES и ITERATE можно использовать для реализации различных методов решения нелинейных уравнений. Так, следующий пример показывает задание функции NEWTON, реализующей метод Ньютона, для решения уравнения  $u(x) = 0$ .

1: "Реализация метода Ньютона"

2: 
$$\text{NEWTON}(u, x, x_0, n) := \text{ITERATE} \left( x - \frac{u}{\frac{d}{dx} u}, x, x_0, n \right)$$

3: NEWTON (x - SIN (x) - 0.25, x, 1.2, 20)

4: 1.17122

Метод Ньютона реализует итерационный процесс по формуле:

$$x_{i+1} = x_i - u(x_i) / u'(x_i)$$

Зависимость шага изменения  $x$  от производной  $u'(x)$  обеспечивает высокую скорость сходимости метода Ньютона.

Другой пример показывает задание функции, решающей систему нелинейных уравнений:

$\text{FIXED\_POINT}(g, x, x_0, n) := \text{ITERATES}(\text{LIM}(g, x, x_k), x_k, x_0, n)$

Здесь  $g$  — вектор правых частей системы нелинейных уравнений вида  $f(x) = x$ ,  $x$  — вектор переменных,  $x_0$  — вектор начальных значений переменных и  $n$  — число переменных.

## 2.6. Векторные и матричные функции

### 2.6.1. Векторы и матрицы

Наряду с отмеченными выше простыми типами данных (целые, вещественные и комплексные числа) Derive использует более сложные типы данных — векторы и матрицы.

Вектор — это просто последовательность чисел

$X_1 X_2 X_3 \dots X_i \dots X_n$

или символьных выражений. Она образует одну строку (или один столбец) данных.

Матрица является совокупностью  $m$  векторов. Далее рассматриваются матрицы с размерностью  $m \times n$ , где  $m$  — число строк матрицы, а  $n$  — число столбцов. Такая матрица может быть представлена в виде:

$$\begin{array}{cccccc} A_{1,1} & A_{1,2} & A_{1,3} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & A_{2,3} & \dots & A_{2,n} \\ \dots & \dots & \dots & A_{j,i} & \dots \\ A_{m,1} & A_{m,2} & A_{m,3} & \dots & A_{m,n} \end{array}$$



Каждый элемент вектора можно представить в виде  $V_j$ , а матрицы (например,  $A$ ) — в виде  $A_{j,i}$ , где  $j$  и  $i$  — индексы, определяющие положение элемента в векторе или матрице. Ниже перечислены основные векторные и матричные функции Derive, причем  $u$  — арифметическое выражение,  $k$  — переменная индекса (цикла),  $v$  и  $w$  — векторы.

## 2.6.2. Векторные функции и операторы

Векторные функции и операторы:

|                                   |                                                                                 |
|-----------------------------------|---------------------------------------------------------------------------------|
| $[x_1, x_2, \dots, x_n]$          | — задание вектора $x$ из $n$ элементов;                                         |
| <b>VECTOR</b> ( $u, k, n$ )       | — задание вектора значений $u$ при $k$ , меняющемся от 1 до $n$ с шагом 1;      |
| <b>VECTOR</b> ( $u, k, m, n$ )    | — задание вектора значений $u$ при $k$ , меняющемся от $m$ до $n$ с шагом 1;    |
| <b>VECTOR</b> ( $u, k, m, n, s$ ) | — задание вектора значений $u$ при $k$ , меняющемся от $m$ до $n$ с шагом $s$ ; |
| <b>ELEMENT</b> ( $v, n$ )         | — выделение $n$ -го элемента вектора $v$ ;                                      |
| <b>DIMENSION</b> ( $v$ )          | — размерность $n$ вектора $v$ ;                                                 |
| $v \cdot w$                       | — скалярное произведение векторов;                                              |
| <b>CROSS</b> ( $v, w$ )           | — векторное произведение двух трехэлементных векторов.                          |

Ниже представлены примеры выполнения основных операций с векторами.

- 1: "Основные операции с векторами"
- 2: "Задание вектора  $X$  с пятью элементами"
- 3:  $x := [1, 2, 3, 4, 5]$
- 4: "Выделение третьего элемента вектора  $X$ "
- 5: **ELEMENT** ( $x, 3$ )
- 6: 3
- 7: "Определение размерности вектора"
- 8: **DIMENSION** ( $x$ )
- 9: 5
- 10: "Действие функций **VECTOR**"
- 11: **VECTOR** ( $k \ y, k, 5$ )
- 12:  $[y, 2 \ y, 3 \ y, 4 \ y, 5 \ y]$
- 13: **VECTOR** ( $k \ x, k, 3$ )
- 14: 
$$\begin{array}{c} / \ 1 \ 2 \ 3 \ 4 \ 5 \ \backslash \\ | \ 2 \ 4 \ 6 \ 8 \ 10 \ | \\ \backslash \ 3 \ 6 \ 9 \ 12 \ 15 \ / \end{array}$$

```

15: VECTOR (k y, k, 3, 6)
16: [3 y, 4 y, 5 y, 6 y]
17: VECTOR (k 2, k, 0, 0.5, 0.1)
18: [0, 0.2, 0.4, 0.6, 0.8, 1]
19: "Задание вектора w"
20: w := [6, 7, 8, 9, 10]
21: "Произведение векторов"
22: x . w
23: 130
24: "Кросс-произведение векторов"
25: CROSS (x, w)
26: CROSS ([1, 2, 3, 4, 5], [6, 7, 8, 9, 10])

```

Прежде всего следует отметить особую роль функции VECTOR, генерирующей и возвращающей векторы. С позиций программирования эта функция вполне заменяет циклы, как целочисленные, так и с нецелочисленной управляющей переменной k. Возможны необычные применения функции VECTOR, когда она используется для задания векторов и матриц, элементы которых являются производными различного порядка, значениями определенных интегралов с разными верхними пределами интегрирования, суммами рядов с разным числом членов и т. д. Ниже приведены примеры таких вычислений.

```

1: "Специальные приемы использования функции VECTOR"
2: "Создание вектора из трех производных функции"
3: Z (x, y) := x2 y3
4: VECTOR (/d \k
 | | -- | Z (x, y), k, 3 |
 \| dx /
5: / \
 | 3 3 |
 | 2 x y , 2 y , 0 |
6: "Создание матрицы из шести производных функции"
7: H (x, y) := x4 y3
8: VECTOR (/ VECTOR (/d \m /d \k
 | | -- | | -- | H (x, y), k, 0, 2 |, m, 1, 2 |
 \| dy / \| dx /

```

$$9: \begin{pmatrix} / & 4 & 2 & & 3 & 2 & & 2 & 2 & \backslash \\ 3 & x & y & & 12 & x & y & & 36 & x & y \\ \backslash & & 4 & & & 3 & & & 2 & & \end{pmatrix}$$

10: "Вычисление определенного интеграла при разных b"

$$11: \text{VECTOR} \begin{pmatrix} / & b & \backslash \\ / & \int \text{SQRT}(2x+1) dx, b, 0.5, 1.5, 0.5 & / \\ \backslash & 0 & \backslash \end{pmatrix}$$

12: [0.609475, 1.39871, 2.33333]

13: "Вычисление сумм членов ряда с разным их числом"

$$14: \text{VECTOR} \begin{pmatrix} / & k & 1 & \backslash \\ \text{SUM} & \frac{1}{n} & k, 100, 500, 100 & / \\ \backslash & n & \backslash \end{pmatrix}$$

15: [5.18737, 5.87803, 6.28266, 6.56993, 6.79282]

Приведенные выше приемы применения функции VECTOR широко используются для вычисления ряда специальных математических функций по их разложениям в ряд. Перейдем к рассмотрению матричных операторов и функций.

### 2.6.3. Матричные функции и операторы

Матричные функции и операторы представлены тем минимумом, который необходим для решения типовых математических задач, связанных с применением матриц. Ниже указаны эти функции и операторы:

|                            |                                                                                         |
|----------------------------|-----------------------------------------------------------------------------------------|
| <b>[[a, b], [c, d]]</b>    | — задание матрицы с размером 2×2;                                                       |
| <b>IDENTITY_MATRIX (n)</b> | — задание единичной матрицы n * n;                                                      |
| <b>ELEMENT (A, j, k)</b>   | — выделение элемента A <sub>j, k</sub> матрицы A;                                       |
| <b>A . B</b>               | — скалярное произведение матриц;                                                        |
| <b>A `</b>                 | — транспонирование матрицы A (замена строк столбцами);                                  |
| <b>DET (A)</b>             | — вычисление детерминанта матрицы A;                                                    |
| <b>TRACE (A)</b>           | — вычисление следа матрицы A (суммы диагональных элементов);                            |
| <b>A ^ -1</b>              | — инвертирование матрицы A;                                                             |
| <b>ROW_REDUCE (A)</b>      | — ступенчатая форма матрицы A;                                                          |
| <b>ROW_REDUCE (A, B)</b>   | — ступенчатая форма матрицы A, расширенная матрицей B;                                  |
| <b>CHARPOLY (A, mu)</b>    | — вычисление характеристического многочлена A с коэффициентами, заносимыми в вектор mu; |
| <b>EIGENVALUES (A, mu)</b> | — вычисление собственных значений матрицы A (заносятся в вектор mu).                    |

Отметим некоторые специальные понятия, относящиеся к матричным вычислениям.

**Сингулярная матрица** — квадратная матрица, у которой детерминант равен 0. Такая матрица не упрощается.

**Единичная матрица** — квадратная матрица, у которой диагональные элементы равны 1, а остальные элементы — это 0.

**Ступенчатая (эрмитова) форма матрицы** соответствует условиям, когда первый ненулевой элемент в каждой строке есть 1, первый ненулевой элемент каждой строки появляется справа от первого ненулевого элемента в предыдущей строке и все элементы выше первого ненулевого в строке — нули.

**Характеристический многочлен матрицы** — определитель разности этой матрицы и единичной матрицы, умноженный на переменную.

**Собственные значения матрицы** — нули ее характеристического полинома. Следующие примеры иллюстрируют применение ряда матричных операций.

1: "Основные операции с матрицами"

2: "Задание матрицы 3\*3"

3: 
$$a := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{pmatrix}$$

4: "Выделение элемента матрицы a2,3"

5: ELEMENT (a, 2, 3)

6: 6

7: "Вычисление детерминанта матрицы a"

8: DET (a)

9: 24

10: "Вычисление следа матрицы a"

11: TRACE (a)

12: 7

13: "Транспонирование матрицы a"

14: a'

15: 
$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 1 \end{pmatrix}$$

16: "Инвертирование матрицы a"

17: 
$$a^{-1}$$

```

18: / -1.79166 0.916666 -0.125 \
 | 1.58333 -0.833333 0.25 |
 \ -0.125 0.25 -0.125 /

```

19: "Создание единичной матрицы 3\*3"

```
20: m := IDENTITY_MATRIX (3)
```

```

21: / 1 0 0 \
 | 0 1 0 |
 \ 0 0 1 /

```

22: "Скалярное произведение матриц"

```
23: a . m
```

```

24: / 1 2 3 \
 | 4 5 6 |
 \ 7 8 1 /

```

25: "Характеристический полином матрицы a"

```
26: CHARPOLY (a, p)
```

```

27: - p3 + 7 p2 + 66 p + 24

```

28: "Собственные значения матрицы a"

```
29: EIGENVALUES (a, v)
```

```
30: [v = -5.07440]
```

31: "Функция ROW\_REDUCE"

```
32: ROW_REDUCE (a)
```

```

33: / 1 0 0 \
 | 0 1 0 |
 \ 0 0 1 /

```

```
34: ROW_REDUCE (a, m)
```

```

35: / 1 0 0 -1.79166 0.916666 -0.125 \
 | 0 1 0 1.58333 -0.833333 0.25 |
 \ 0 0 1 -0.125 0.25 -0.125 /

```

Для получения ступенчатой формы несингулярной матрицы  $A$  используется функция  $\text{ROW\_REDUCE}(A)$ . Функция  $\text{ROW\_REDUCE}(A, B)$  соединяет две матрицы и возвращает ступенчатую форму для расширенной матрицы.

Эти функции можно использовать для решения матричных уравнений вида

$$A X = B.$$

Например, если  $A$  и  $B$  — матрицы

$$\begin{vmatrix} 1 & 2 \\ 5 & 6 \end{vmatrix} \quad \text{и} \quad \begin{vmatrix} 3 & 4 \\ 7 & 8 \end{vmatrix}$$

то применение  $\text{ROW\_REDUCE}(A, B)$  упрощается к матрице

$$\begin{vmatrix} 1 & 0 & -1 & -2 \\ 0 & 1 & 2 & 3 \end{vmatrix}$$

т. е. решение есть

$$\begin{vmatrix} -1 & -2 \\ 2 & 3 \end{vmatrix}.$$

#### 2.6.4. Матричные операции в символьной форме

При упрощении и нормализации выражений, содержащих векторные и матричные операторы,  $\text{Derive}$  использует следующие соотношения векторной и матричной алгебры:

$$a'' \rightarrow a$$

$$[1/a]' \rightarrow \frac{1}{a'}$$

$$(a+b)' \rightarrow a' + b'$$

$$(a \cdot b)' \rightarrow b' \cdot a'$$

$$a \cdot (b+c) \rightarrow a \cdot b + a \cdot c$$

$$(b+c) \cdot a \rightarrow b \cdot a + c \cdot a$$

$$(a \cdot b) \cdot c \rightarrow a \cdot (b \cdot c)$$

$$\frac{1}{a \cdot b} \rightarrow \frac{1}{b} \cdot \frac{1}{a}$$

$$\text{DET} [1/a] \rightarrow \frac{1}{\text{DET } a}$$

$$\text{DET} (a \cdot b) \rightarrow \text{DET} (a) \text{DET} (b)$$

В ряде случаев большой интерес представляет возможность выполнения матричных операций в символьном виде. Примеры с такими операциями приведены ниже.

1: "Символьные операции с матрицами"

2: "Задание матрицы в символьном виде"

$$3: m := \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

4: "Транспонирование матрицы"

5:  $m'$

$$6: \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

7: "Инвертирование матрицы"

8:  $m^{-1}$

$$9: \begin{pmatrix} d & b \\ a d - b c & a d - b c \\ c & a \\ a d - b c & a d - b c \end{pmatrix}$$

10: "Задание вектора"

$$11: v := \begin{pmatrix} e \\ f \end{pmatrix}$$

12: "Решение системы линейных уравнений  $m \cdot x = v$ "

13: "Используйте далее команду Simplify"

$$14: x := \frac{v}{m}$$

$$15: \begin{pmatrix} e \\ f \end{pmatrix} \begin{pmatrix} d & b \\ a d - b c & a d - b c \\ c & a \\ a d - b c & a d - b c \end{pmatrix}$$

16: "Вычисление детерминанта матрицы  $m$ "

17:  $\text{DET}(m)$

18:  $a d - b c$

19: "Вычисление следа матрицы  $m$ "

20: TRACE (m)

21: a + d

22: "Задание матрицы с элементами-выражениями"

$$23: \text{ms} := \begin{pmatrix} / & 1 & & 2 z & \backslash \\ & \text{-----} & & \text{-----} & \\ & 2 & & 2 & \\ & z - 3 & & z - 3 & \\ & \text{-----} & & \text{-----} & \\ & 7 & & 6 & \\ & \text{-----} & & \text{-----} & \\ \backslash & z - 2 & & z - 5 & / \end{pmatrix}$$

24: "Вычисление детерминанта матрицы ms"

25: DET (ms)

26: "Используйте команду Simplify"

$$27: - \frac{2 (7 z^2 - 38 z + 6)}{(z - 3) (z - 2) (z - 5)}$$

Следует отметить, что элементами матриц и векторов могут быть арифметические выражения. Тогда символьные операции выполняются над ними (см. пример в строках 22–27). Результаты операций могут быть представлены в матричной форме (см. пример на решение системы линейных уравнений в строках 12–15). К сожалению, символьные операции возможны лишь с матрицами малого размера. В противном случае символьные представления результатов оказываются настолько громоздкими, что практическая польза от них становится сомнительной. Для работы с ними их лучше распечатать на принтере, чем созерцать на экране дисплея.

У ранних версий Derive при работе на ПК с ОЗУ емкостью 640 Кбайт объем памяти не позволял работать с матрицами, имеющими более 25 ненулевых элементов. В версии Derive 3.11 максимальный размер матриц существенно выше и зависит от емкости ОЗУ.

## 2.7. Статистические и финансовые функции

Derive включает достаточный набор функций для проведения статистических расчетов и типовой статистической обработки данных. Прежде всего отметим четыре группы функции, широко применяемые в теоретической статистике.

### 2.7.1. Функции ошибок

Функции ошибок:

ERF (x)

— функция ошибок;

ERF (x, y)

— функция ошибок общего вида;



- ERFC (x)** — дополнительная функция ошибок;  
**NORMAL (x, m, s)** — функция нормального распределения со средним  $m$  и среднеквадратичным  $s$ ;  
**NORMAL (z)** — интегральное распределение по  $z$ .

Эти функции с равным успехом можно было бы отнести к специальным математическим функциям. Они возвращают единственное значение и имеют аргументы скалярного типа. Поясним следующими примерами:

- 1: "Вычисление функций ошибки"
- 2: ERF (0.5)
- 3: 0.520499
- 4: ERF (0.5, 2)
- 5: 0.474822
- 6: "Вычисление дополнительной функции ошибок"
- 7: ERFC (0.5)
- 8: 0.479500
- 9: "Проверка равенства  $\text{erf}(x) + \text{erfc}(x) = 1$ "
- 10: ERF (x) + ERFC (x)
- 11: 1
- 12: "Вычисление функции нормального распределения"
- 13: NORMAL (0.5, 1, 1)
- 14: 0.308537

Довольно широкая группа статистических функций обеспечивает вычисление статистических параметров групповых данных. Данные при этом задаются в виде векторов или просто перечислением. Ниже представлены основные из этих функций.

## 2.7.2. Основные статистические функции

Статистические функции:

- AVERAGE (z1, ..., zn)** — среднее арифметическое;  
**RMS (z1, ..., zn)** — среднеквадратичное значение;  
**VAR (z1, ..., zn)** — дисперсия;  
**STDEV (z1, ..., zn)** — среднеквадратичное отклонение;  
**FIT (m)** — обработка матрицы  $m$  методом наименьших квадратов;  
**FIT (v, A)** — многомерная линейная регрессия вектора  $v$ , задающего аргументы и функцию по матрице данных  $A$ .

На представленных ниже примерах иллюстрируется действие большинства этих функций.

- 1: "Статистические функции"
- 2: "Используйте команду Simplify"
- 3: AVERAGE (a, b, c)
- 4: 
$$\frac{a + b + c}{3}$$
- 5: AVERAGE (1, 2, 3)
- 6: 2
- 7: RMS (a, b)
- 8: 
$$\frac{\text{SQRT}(2) \text{SQRT}(a^2 + b^2)}{2}$$
- 9: VAR (a, b)
- 10: 
$$\frac{(a - b)^2}{4}$$
- 11: RMS (1, 1.1, 1.2, 1.3, 1.4)
- 12: 
$$\frac{\text{SQRT}(146)}{10}$$
- 13: VAR (1, 1.1, 1.2, 1.3, 1.4)
- 14: 
$$\frac{1}{50}$$

Обширными возможностями статистической обработки данных обладает функция FIT. Она будет рассмотрена отдельно чуть позднее.

### 2.7.3. Финансово-экономические функции

Для проведения финансово-экономических расчетов в Derive включен ряд функций. Все они ориентированы на расчет вкладов и базируются на применении формулы

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

со следующими параметрами:

- i — процентная ставка;
- v (pval) — настоящее (во время проведения расчетов) значение (сумма) вклада;

- f (fval) — будущее значение (сумма) вклада;  
 n (nper) — число регулярных вложений;  
 t (time) — время платежа (необязательный аргумент);  
 p (pmt) — пай (вложение или изъятие).

Аргумент t (time) означает время платежа. Он задается так:

- 0 — если платежи производятся в конце каждого периода;  
 1 — если платежи производятся в начале каждого периода;  
 пропорциональная дробь в промежутке от 0 до 1 — если платежи производятся внутри каждого периода.

По умолчанию задается значение t (time), равное 0. Приведенная формула весьма чувствительна к погрешностям численных расчетов. Заданная по умолчанию точность в 6 значащих цифр обычно недостаточна для корректных вычислений и рекомендуется задать не менее 10 значащих цифр.

Финансово-экономические функции Derive возвращают:

- PVAL (i, nper, pmt, fval, time) — приведенную стоимость контракта;  
 FVAL (i, nper, pmt, pval, time) — будущую стоимость контракта;  
 PMT (i, nper, pval, fval, time) — периодические платежи;  
 NPER (i, pmt, pval, fval, time) — число платежных периодов;  
 RATE (nper, pmt, pval, fval, time, min, max) — периодическую процентную ставку.

В отношении знаков финансовых величин в Derive действует правило: знак минус означает, что пользователь отдает (вносит) некоторую сумму, а знак плюс — что он ее получает. Несмотря на логичность этого правила, надо отметить, что в финансовой литературе оно нередко не используется. Поэтому вид расчетных формул у Derive может иметь отличия от принятых, легко выявляемые, если учесть это правило. Кроме того, интересно, что можно использовать упрощенные выражения, содержащие меньшее число параметров, чем указано выше (недостающие Derive задает по смыслу вычислений).

Ниже представлено решение типовых задач с вкладами.

- 1: "Финансово-экономические расчеты"  
 2: "Вкладчик ежемесячно на протяжении 10 лет отдает по"  
 3: "250 руб. Какую сумму при 5.25% годовых он накопит?"  
 4: FVAL (i, n, p)

$$5: \frac{p}{i} - \frac{p(1+i)^n}{i}$$

$$6: \text{FVAL} \left( \frac{5.25\%}{12}, 12 \cdot 10, -250 \right)$$

7: 39344

8: "Сколько раз вкладчик должен вкладывать по 250 руб."

9: "при 5.25% годовых, чтобы накопить 39344 руб.?"

10: NPER (i, p, f)

$$11: \frac{\text{LN} \left( \frac{p}{f i + p} \right)}{\text{LN} (i + 1)}$$

$$12: \text{NPER} \left( \frac{5.25\%}{12}, 250, 39344 \right)$$

13: -120

14: "Какую сумму надо вложить, чтобы при ежегодных"

15: "вкладах в 1000 руб. при 10% годовых накопить"

16: "через 10 лет сумму в 50000 руб.?"

17: PVAL (10%, 10, -1000, 50000, 10)

18: -6988

19: "Какая сумма накопится за 10 лет при ежегодных"

20: "вложениях по 1000 руб., начальном вкладе"

21: "7000 руб. и проценте годовых 10%?"

22: FVAL (10%, 10, -1000, -7000, 10)

23: 50031

В этих примерах показано, что результаты расчетов можно получить как в символьном виде (если они представимы в виде формулы), так и в числовом. Рекомендуется внимательно протестировать необходимые расчеты, прежде чем полагаться на их полную достоверность. Тут и пригодится вывод формул финансовых расчетов и сравнение их с формулами, которые используются в отчетах или литературе пользователя.

## 2.8. Реализация метода наименьших квадратов

### 2.8.1. Функция FIT

Особая роль принадлежит отмеченной выше функции FIT(m). Кратко говоря, ее назначение заключается в обработке матрицы m методом наименьших квадратов. Это емкое определение означает возможность решения задач на регрессию различных зависимостей — от простейшей линейной регрессии до линейной и нелинейной регрессии общего вида. Задачи такого типа широко распространены в физических и научно-технических приложениях статистических методов и поэтому требуют достаточно подробного обсуждения.

### 2.8.2. Линейная регрессия

Начнем с простейшей и одной из наиболее распространенных задачи — линейной регрессии. Пусть надо облако точек (x, y) представить линейной зависимостью

$$y(x) = a \cdot x + b$$

При этом требуется найти параметры  $a$  и  $b$ , при которых среднеквадратичная погрешность приближения минимальна.

Для решения задачи с помощью функции  $FIT(m)$  надо создать матрицу  $m$ , имеющую два столбца. Верхняя строка матрицы должна содержать переменную  $x$  и функцию  $y(x)$ , в нашем случае — это  $a \cdot x + b$ . Остальные строки должны содержать пары значений  $x$  и  $y$ , т. е. исходные данные (например, экспериментальные). Вычисление функции  $FIT(m)$  с помощью команды `арргоХ` даст искомую зависимость. Следующий пример поясняет это:

1: "Реализация метода наименьших квадратов"

2: "Линейная регрессия"

3: FIT 
$$\begin{array}{|c|} \hline / \quad x \quad a \cdot x + b \quad \backslash \\ \hline | \quad 2 \quad 5.5 \\ \hline | \quad 4 \quad 6.3 \\ \hline | \quad 6 \quad 7.2 \\ \hline | \quad 10 \quad 8.6 \\ \hline | \quad 8 \quad 8 \\ \hline \backslash \end{array}$$

4: "Используем команду `арргоХ`"

5:  $0.395 x + 4.75$

Итак, в приведенном примере облако точек исходной зависимости приближено функцией  $0,369x + 4,75$ . При этом обеспечивается наименьшая среднеквадратичная погрешность приближения этой функцией, которая является прямой.

### 2.8.3. Логарифмическая регрессия

Столь же просто можно провести логарифмическую регрессию.

1: "Реализация метода наименьших квадратов"

2: "Логарифмическая регрессия"

3: FIT 
$$\begin{array}{|c|} \hline / \quad x \quad a + b \cdot \text{LOG}(x, 10) \quad \backslash \\ \hline | \quad 1 \quad 1 \\ \hline | \quad 2 \quad 1.45 \\ \hline | \quad 3 \quad 1.7 \\ \hline | \quad 4 \quad 1.9 \\ \hline \backslash \end{array}$$

4:  $0.645501 \text{ LN}(x) + 0.999640$

Облако точек тут приближено логарифмической функцией  $0,645501 \ln(x) + 0,99964$ .

### 2.8.4. Полиномиальная регрессия

Другой пример показывает использование функции FIT(m) для полиномиальной регрессии.

1: "Реализация метода наименьших квадратов"

2: "Полиномиальная регрессия"

3: FIT 
$$\left( \begin{array}{c} x^3 + a x^2 + b x + c \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \quad \begin{array}{c} x^2 + d \\ 1 \\ 4 \\ 8.5 \\ 8 \end{array} \right)$$

4:  $-1.08333 x^3 + 7.25 x^2 - 11.1666 x + 6$

В этом примере облако точек приближено полиномом третьей степени, представленным в строке #4.

### 2.8.5. Линейная регрессия общего вида

Универсальность заложенных в реализацию функции FIT(m) алгоритмов позволяет проводить и более сложные вычисления. Так, можно осуществить линейную регрессию общего вида, когда отыскиваются коэффициенты  $a, b, c, \dots$ , обеспечивающие приближение облака исходных точек комбинацией функций:

$$y(x) = a*f_1(x) + b*f_2(x) + c*f_3(x) + \dots$$

с наименьшей среднеквадратичной погрешностью.

Ниже представлен пример для зависимости

$$y(x) = a*x^2 + b*e^{-x} + c/x$$

1: "Реализация метода наименьших квадратов"

2: "Линейная регрессия общего вида"

3: FIT 
$$\left( \begin{array}{c} x^2 + a x + b \text{ EXP}(x) + \frac{c}{x} \\ 1 \\ 2 \\ -0.5 \\ -10 \\ -1 \end{array} \quad \begin{array}{c} 4.73 \\ 14.5 \\ -1.15 \\ 100 \\ 0.368 \end{array} \right)$$

$$4: \quad 1.31098 \#e^x + 1.00101 x^2 + \frac{0.980392}{x}$$

Здесь надо обратить внимание на то, что выбранная функция регрессии должна быть линейна относительно искомым параметров, но может быть нелинейной относительно переменной  $x$ .

## 2.8.6. Регрессия для функции нескольких переменных

Функцию FIT( $m$ ) можно использовать и для приближения по методу наименьших квадратов функций ряда переменных. При этом соответственно увеличивается число столбцов матрицы  $m$ . Пусть данные надо описать зависимостью

$$z(x, y) = a * x + b * \exp(y) + c.$$

Это можно сделать следующим образом.

- 1: "Реализация метода наименьших квадратов"
- 2: "Функция двух переменных"

$$3: \quad \text{FIT} \left( \begin{array}{c|cc|cc|c} & x & y & a x + b & \text{EXP}(y) + c & \\ \hline & 1 & 1 & & 9.43 & \\ & 2 & 1 & & 10.4 & \\ & 1 & 2 & & 18.8 & \\ & 2 & 2 & & 19.8 & \end{array} \right)$$

$$4: \quad 2.00930 \#e^y + 0.985000 x + 2.97562$$

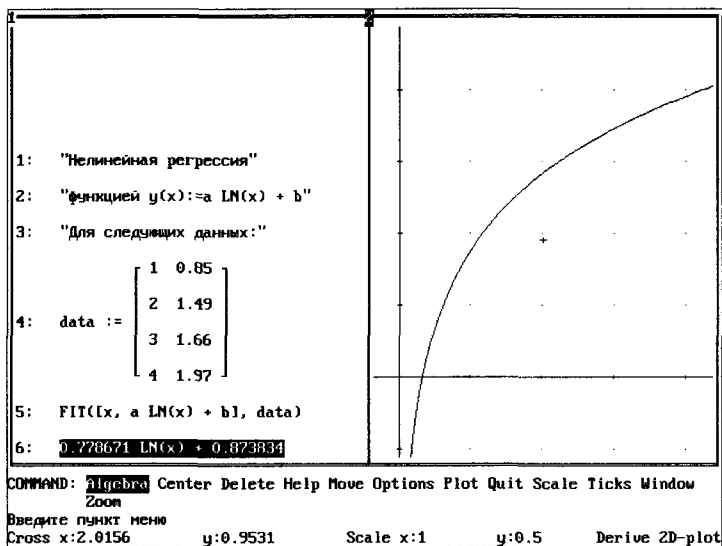
Не следует думать, что Derive непременно повторяет заданное уравнение регрессии. В действительности система пытается провести возможные аналитические преобразования этого уравнения, в чем можно убедиться при внимательном рассмотрении результатов логарифмической регрессии, когда система изменила уравнение регрессии, заменив десятичный логарифм натуральным. В некоторых случаях результаты регрессии могут оказаться весьма неожиданными, если уравнение регрессии допускает различные и далеко идущие преобразования.

## 2.8.7. Нелинейная регрессия с выводом данных в графической форме

При решении задач регрессии и аппроксимации весьма полезно сопровождать их выводом результатов в графической форме. Наглядность задач намного возрастает, если наряду с построением графика функции регрессии строятся узловые точки и непосредственно видна реализация алгоритма решения задачи.

На рис. 2.1 показано обычное решение задачи на нелинейную регрессию. После ввода первых пяти строк с применением команды Author выполняется команда arпроX и получается уравнение регрессии — строка 6. Затем окно де-

**Рис. 2.1**  
Выполнение  
регрессии  
с построением  
графика ее  
функции

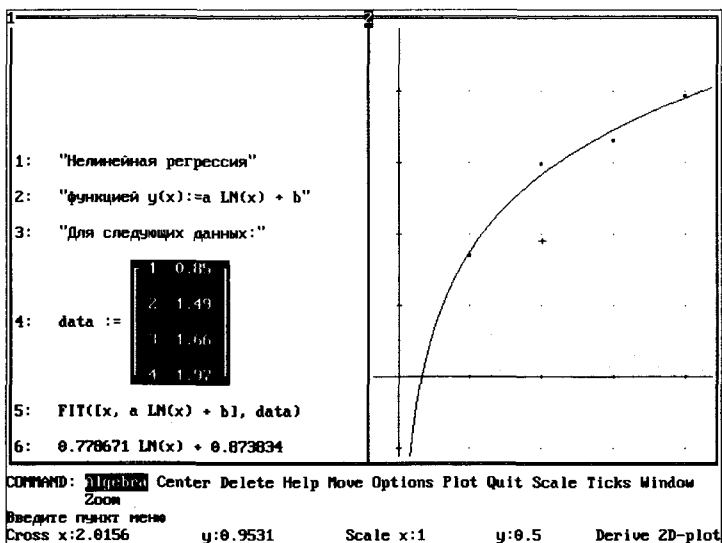


лится по вертикали и во втором окне (с использованием команды Plot) строится график функции регрессии.

Для построения узловых точек в окне графика нужно перейти вначале в окно с выражениями и выделить матрицу исходных данных. Затем, перейдя в окно графики, следует исполнить команду Plot — будут построены узловые точки с цветом, отличающимся от цвета линии регрессии. Это показано на рис. 2.2 (разумеется, без различия цветов линии регрессии и узловых точек).

Рис. 2.2 очень наглядно представляет результат регрессии. Хорошо видно, что линия регрессии проходит в максимальной близости от узловых точек, но не совпадает с ними точно (как это бывает при проведении полиномиальной аппроксимации). Результаты регрессии можно распечатать на

**Рис. 2.2**  
Окно Derive после  
проведения  
регрессии и  
построения  
узловых точек  
(слева выделена  
матрица с  
координатами  
узловых точек)





принтере или направить в файл с расширением .tif. Причем эти операции возможны как со всем экраном, так и с отдельными окнами.

Возможно также копирование экрана внешними программами (например, резидентной capture) с созданием файла с расширением .hcx и его последующим редактированием с помощью подходящего графического редактора, например Microsoft PaintBrush. Такое редактирование полезно, в частности, для обозначения кривых или узловых точек, нанесения цифр делений масштабных осей и т. д.

## 2.9. Функции градиентного и векторного анализа

Возможности системы Derive расширены элементами градиентного и векторного анализа, что позволяет решать довольно широкий класс задач теории поля (электромагнитного, теплового, гравитационного и пр.). Derive позволяет вычислять вектор-градиент для функции  $n$ - переменных, представленной выражением  $\text{expr}$ . Тут необходимо указать эти переменные, как элементы некоторого вектора  $v$ . Ниже представлены функции и операторы для вычисления параметров дифференциального вектора.

**GRAD (expr)** — градиент  $\text{expr}$  в системе координат с переменными  $x, y$  и  $z$ ;

**GRAD (expr, v)** — градиент  $\text{expr}$  в системе координат с переменными из вектора  $v$ ;

**GRAD (expr, A)** — градиент  $\text{expr}$  в системе координат  $A$ ;

**DIV (v, A)** — дивергенция вектора  $v$ ;

**LAPLACIAN (expr, A)** — дивергенция градиента  $\text{expr}$ ;

**CURL (v, A)** — ротор вектора  $v$ .

Ряд функций Derive позволяет вычислять потенциалы (параметры интегрального вектора):

**POTENTIAL (v)** — скалярный потенциал вектора;

**POTENTIAL (v, w)** — потенциал вектора с начальными координатами  $w$ ;

**POTENTIAL (v, w, A)** — потенциал вектора в системе координат  $A$ ;

**VECTOR\_POTENTIAL (v, w, A)** — векторный потенциал вектора.

Ниже представлены задачи на применение приведенных выше функций (для вычислений используется команда Simplify).

1: "Функции дифференциальных и интегральных векторов"

2:  $\text{GRAD}(x^3 + y^2 + z)$

3:  $\sqrt[3]{x^2}, 2y, 1$

4:  $\text{GRAD}(x^3 + y^2 + z, [x, y, z])$

5:  $\sqrt[3]{x^2}, 2y, 1$

$$6: \text{DIV} \left( \sqrt[3]{1, 2 y, 4 z^2} \right)$$

$$7: 8 z + 2$$

$$8: \text{LAPLACIAN} (x^3 + y^2 + z)$$

$$9: 6 x + 2$$

$$10: \text{CURL} \left( \sqrt[3]{y, 2 x^2 z, 1} \right)$$

$$11: \left( -2 x^2, 0, 4 x z - 3 y^2 \right)$$

$$12: \text{POTENTIAL} \left( \sqrt[3]{x, 2 y^2, 3 z^3} \right)$$

$$13: \frac{x^2}{2} + \frac{2 y^3}{3} + \frac{3 z^4}{4}$$

$$14: \text{VECTOR\_POTENTIAL} \left( \sqrt[3]{x, 2 y^2, 3 z^3} \right)$$

$$15: \left( \sqrt[3]{2 y^2 z, -x z, 0} \right)$$

Описанные здесь функции свидетельствуют об обширных возможностях системы Derive. Они могут быть дополнительно расширены с помощью библиотек функций. Им посвящена следующая глава.

## 2.10. Примеры и специальные приемы программирования

### 2.10.1. Пример вычисления модульной теплоемкости по Дебаю

Выше мы рассмотрели синтаксис Derive и полный набор встроенных операторов и функций системы. Однако само программирование рассматривалось на крайне примитивном уровне — на примерах работы отдельных функций. В этом же разделе приводится ряд дополнительных примеров, демонстрирующих более развитые возможности Derive.

Существует множество задач, где функции Derive могут применяться прямо, без каких-либо специальных ухищрений. Рассмотрим, к примеру, вычисление мольной теплоемкости металлов по Дебаю. Это обычная интегральная функция, вычисление которой показано ниже.

1: "Вычисление мольной теплоемкости металлов по Дебаю"

2: "Z=@/T отношение температуры Дебая @ к температуре T"

$$3: \quad CV(z) := \frac{9}{3} r \int_0^z \frac{x^4 \text{EXP}(x)}{(\text{EXP}(x) - 1)^2} dx$$

4: "Используйте команду approx"

5: CV(1)

6: 2.85519 r

7: CV(3)

8: 1.98827 r

Мольная теплоемкость выражается функцией  $CV(z)$ , где  $z$  есть отношение температуры Дебая @ к абсолютной температуре  $T$ . Результат представляется как произведение числового коэффициента на универсальную газовую постоянную  $r$ .

## 2.10.2. Функции с параметрами-функциями

Однако во многих случаях желательно, чтобы созданная пользователем функция могла иметь в качестве параметра не просто вычисляемое тут же арифметическое выражение, а произвольно заданную функцию. Derive позволяет передавать символическое значение функций, введенных в качестве параметров в лист параметров, если эти функции используются в составе встроенных функций Derive. Эта формулировка становится ясной из следующего простого примера.

1: "Функция пользователя с параметром-функцией"

2:  $F(y, x, a) := \text{LOG}(\lim_{x \rightarrow a} y)$

3:  $F(x^2, x, 2)$

4: 1.38629

5: "Проверка"

6:  $\text{LOG}(2^2)$

7: 1.38629

8:  $F(x^3, x, 2)$

9: 2.07944

10: "Проверка"

11:  $\text{LOG} (2^3)$

12: 2.07944

В этом простом, но отнюдь не тривиальном примере задана некоторая функция  $F(y, x, a)$ . Однако здесь  $y$  и  $x$  — не обычные параметры. Они позволяют передать в правую часть выражения #2 некоторую произвольную функцию  $y(x)$ , указанную обобщенным именем  $y$  и имеющую аргумент  $x$ . А вот  $a$  — обычный параметр. Приведенная функция вычисляет натуральный логарифм для любого значения  $y(x)$  при  $x = a$ . Для этого несколько необычно использована функция вычисления предела — здесь она просто вычисляет значение функции  $y(x)$  в точке  $a$ . Такой прием связан с тем, что прямо использовать функцию  $y(x)$  в правой части выражения #2 нельзя, в ней могут стоять лишь определенные ранее (т. е. встроенные) функции. В нашем примере параметры  $y$  и  $x$  передают в тело функции  $F(y, x, a)$  вначале функцию  $x^2$ , а затем функцию  $x^3$ .

### 2.10.3. Задание функции для вычисления интеграла по формуле Уэддла

Весьма полезным может оказаться другой пример — вычисление определенного интеграла вида

$$\text{INT\_U}(f(x), x, a, h) = \int_a^{a+h} f(x) dx$$

по формуле Уэддла [1, 2], дающей высокую точность при разбивке интервала интегрирования от  $a$  до  $b = a + h$  на шесть частей. Вычисления при этом могут происходить заметно быстрее, чем по встроенной функции вычисления интегралов.

Ниже показана реализация этого примера.

1: "Вычисление определенного интеграла по формуле Уэддла"

2:  $\text{INT\_U}(y, x, a, h) := \frac{3}{10} h \left( \lim_{x \rightarrow a} y + 5 \lim_{x \rightarrow a+h} y + \lim_{x \rightarrow a+2h} y \right)$

$+ 6 \lim_{x \rightarrow a+3h} y + \lim_{x \rightarrow a+4h} y + 5 \lim_{x \rightarrow a+5h} y + \lim_{x \rightarrow a+6h} y$

3:  $\text{INT\_U} \left( \text{SQRT}(2x+1), x, 0, \frac{1}{6} \right)$

4: 1.39871

5: "Проверка"

$$6: \int_0^1 \text{SQRT}(2x + 1) dx$$

7: 1.39871

### 2.10.4. Минимизация функций

Derive не содержит встроенных функций минимизации функции одной или многих переменных. Пожалуй, это единственный серьезный недостаток системы, учитывая ту огромную роль, которую задачи минимизации (или, точнее, оптимизации) играют в инженерном проектировании. Разумеется, простые задачи оптимизации непрерывной функции одной переменной легко могут быть сведены к решению уравнения на поиск нулевого значения производной функции.

1: "Вычисление максимума функции:"

2:  $F(x) := x \text{ EXP}(-x)$

3:  $x_{\max} := \text{SOLVE} \left( \begin{array}{l} /d \\ -- F(x), x \\ \backslash dx \end{array} \right)$

4:  $[x = 1]$

5:  $F(0.99)$

6: 0.367860

7:  $F(1)$

8: 0.367879

9:  $F(1.01)$

10: 0.367861

В строках 5–10 приведена простейшая проверка, показывающая, что значение  $x = 1$  здесь действительно есть значение, при котором функция  $F(x)$  достигает максимума, а не минимума, поскольку по обе стороны от значения  $x = 1$  величина  $F(x)$  уменьшается.

Учитывая возможности задания функций с параметрами — произвольными функциями, можно создать более совершенную функцию для поиска экстремумов (как максимумов, так и минимумов) в заданном отрезке изменения аргумента  $[a, b]$ :

1: "Поиск экстремумов функции одной переменной"

2:  $X\_EXTR(y, x, a, b) := \text{SOLVE} \left( \begin{array}{l} /d \\ -- y, x, a, b \\ \backslash dx \end{array} \right)$

3: X\_EXTR (x EXP (-x), x, 0.5, 1.5)

4: [x = 1]

5: X\_EXTR  $\left( \frac{\text{SIN}(x)}{x}, x, -1, 1 \right)$

6: [x = 0]

7: X\_EXTR (SIN (x), x, -2, -1)

8: [x = -1.57080]

В этих примерах показано вычисление максимума для функций  $x \cdot \exp(-x)$  и  $\sin(x)/x$ , а также минимума функции  $\sin(x)$  вблизи точки  $x = -\pi/2$ . Задание отрезка  $[a, b]$  поиска экстремума облегчает анализ многоэкстремальных функций.

Поиск экстремумов функций многих переменных — одна из наиболее сложных вычислительных задач. Хотя ее можно решать средствами Derive, вряд ли это целесообразно, поскольку специализированные программы минимизации функций делают это быстрее и эффективнее.

## 2.10.5. Организация циклов и условных выражений

При программировании на входном языке Derive надо учитывать, что этот язык не имеет общепринятых управляющих структур, таких, как циклы или безусловные и условные переходы. Это язык, ориентированный на использование и создание функций. Тем не менее функции циклов в большинстве случаев заменяет функция VECTOR, а также функции SUM и PROD (в обычных программах именно для их реализации чаще всего и используются циклы).

Следующий пример иллюстрирует применение функции SUM для решения типовой математической задачи — вычисления интегральной показательной функции  $Ei(x)$  путем суммирования  $m$  членов разложения ее в ряд.

1: "Вычисление функции  $Ei(m, x)$ "

2: ce := 0.577215

3:  $EI(m, x) := ce + LN(x) + \sum_{k=1}^m \frac{x^k}{k!}$

4: EI(10, 0.5)

5: 0.454219

В терминах Derive функция задана как  $Ei(m, x)$ . Derive не содержит явных средств для вычислений сумм или произведений сходящихся рядов при неограниченном числе их членов. Поэтому проще всего заменять такие ряды рядами с конечным и достаточно большим числом членов. Так, в приведенном примере при  $m = 10$  точными являются все цифры результата для  $Ei(0.5)$ .

Нет в Derive также средств для организации условных и безусловных переходов (хотя сохранился Бейсик — подобный “анахронизм” в виде пронумерованных строк, кстати, весьма удобен для их поиска и идентификации). Что касается переходов по строкам (особенно безусловных), то они давно признаны лишь злейшим врагом структурного программирования. Стало быть, остается лишь сожалеть, что в Derive нет условных выражений для выбора нужных функций. К счастью, это не совсем так. С помощью функции IF, а также функций STEP и CHI можно осуществлять условный выбор нужных функций.

## 2.10.6. Вычисление коэффициентов Берга

Проиллюстрируем это практическим примером из радиотехники. Пусть требуется вычислить относительную амплитуду  $n$ -й гармоники для отрезка синусоиды с углом отсечки  $\theta$  (или проще  $t$ ). Тут пригодны три разные формулы для коэффициентов Берга, дающие относительную амплитуду при  $n$ , равных 0, 1 и  $n \geq 2$ . Ниже показано конструирование функции BERG( $n, t$ ), вычисляющей коэффициенты Берга при любых целых  $n$  и углах  $t$  (в радианах).

1: "Вычисление коэффициентов Берга"

$$2: A0(t) := \frac{\text{SIN}(t) - t \text{COS}(t)}{\text{pi}(1 - \text{COS}(t))}$$

$$3: A1(t) := \frac{t - \text{SIN}(t) \text{COS}(t)}{\text{pi}(1 - \text{COS}(t))}$$

$$4: AN(n, t) := \frac{\text{pi}^2 (\text{SIN}(n t) \text{COS}(t) - n \text{COS}(n t) \text{SIN}(t))}{n(n^2 - 1)(1 - \text{COS}(t))}$$

$$5: A01(n, t) := \text{IF}(n > 0.5, A1(t), A0(t))$$

$$6: \text{BERG}(n, t) := \text{IF}(n > 1.5, AN(n, t), A01(n, t))$$

$$7: \text{BERG}\left(0, \frac{\text{pi}}{6}\right)$$

8: 0.110598

$$9: \text{BERG}\left(1, \frac{\text{pi}}{6}\right)$$

10: 0.215223

$$11: \text{BERG}\left(2, \frac{\text{pi}}{2}\right)$$

12: 0.212206

В этих примерах в строках 2–4 заданы три исходные функции A0, A1 и AN, вычисляющие коэффициенты Берга для  $n = 0, 1$  и любых  $n$ , равных или больших 2. В строке 5 определена функция A01, выбирающая с помощью функции IF функцию A0 при  $n = 0$  или A1 при  $n = 1$ . Далее в строке 6 определена искомая функция BERG( $n, t$ ), которая также с помощью функции IF выбирает функцию A01, если  $n$  меньше 2, и AN, если  $n$  больше или равно 2. Таким образом, основная функция BERG( $n, t$ ) вычисляет коэффициенты Берга при любых целых  $n$  и углах  $t$  – в строках 7–12 приведены примеры этого.

### 2.10.7. Функции выделения различных частей отношения

Некоторые системы компьютерной алгебры обладают богатым набором функций для работы с частями математических выражений. У Derive таких возможностей нет. Впрочем, начиная с версии 2.60 появились две функции такого рода:

**LHS (r)** – возвращает левую часть отношения  $r$ ;

**RHS (r)** – возвращает правую часть отношения  $r$ .

Они могут быть полезны для программирования и решения некоторых задач символьной математики.

### 2.10.8. Несколько слов о рекурсии

Реализация Derive на языке muLISP позволяет в полной мере реализовать возможности рекурсии путем задания рекурсивных функций, которые могут обращаться к самим себе. Наглядным примером служит рекурсивная функция задания факториала:

$$\text{FACT}(n) := \text{IF}(n=0, 1, n \text{ FACT}(n-1))$$

Она соответствует рекурсивному определению факториала:

$$n! = 1 \quad \text{если } n = 0$$

$$n! = n(n-1)! \quad \text{если } n > 0$$

и содержит внутри себя обращение к самой себе.

При задании рекурсивных функций следует не забывать об условиях их завершения. Обычно они создаются с помощью условных функций IF. К примеру, в функции FACT( $n - 1$ ) значение  $n$  при каждой рекурсии уменьшается на 1, пока не станет равным нулю. В итоге работа функции прекращается.

Возможна также взаимная рекурсия. При ней одна функция обращается к другой, та к третьей и т. д., причем последующие функции могут обращаться к первой функции. Глубокая рекурсия и взаимная рекурсия требуют больших затрат памяти.

Далеко не всегда рекурсивное задание функций оптимально и дает максимальную производительность при вычислениях. Чаще всего все обстоит иначе. Тем не менее рекурсия – основополагающий прием программирования. Следует отметить, что сам по себе язык muLISP базируется на множестве функций, реализующих взаимную рекурсию.



## 2.11. Заключительные замечания по программированию

Ограничимся этими примерами функционального программирования на входном языке Derive как достаточно типовыми. Следует отметить, что прилагаемая к Derive библиотека функций, описанных в следующей главе, является множеством прекрасно подобранных примеров практического программирования в среде Derive. Они наглядно показывают, что с помощью весьма ограниченных средств задания управляющих структур Derive способна решать довольно сложные математические задачи — от вычисления численными методами различных специальных функций до решения систем дифференциальных уравнений современными численными и аналитическими методами.

Завершая главу, посвященную программированию, стоит отметить одно важное обстоятельство — соглашение о назначении идентификаторов внутри функций, заменяющих управляющие структуры, например SUM, PRODUCT, VECTOR, ITERATES и ITERATE и др. Такие идентификаторы нежелательно назначать обычным образом, т. е. в виде букв a, b, ... z. Дело в том, что изменение их значений внутри этих функций является для пользователя неожиданным и нежелательным сюрпризом, способным резко нарушить, казалось бы, верный алгоритм работы программы.

Для избежания таких эффектов в системе Derive принято обозначать внутренние и служебные переменные со знаком “\_” после букв, например a\_, b\_, ..., z\_. Разумеется, надо взять за правило не вводить этот знак в идентификаторы обычных глобальных переменных. Тогда вы будете застрахованы от такой неожиданности, как непредусмотренное изменение значений переменных при исполнении той или иной функции.

# Глава 3. Описание библиотек внешних расширений

## 3.1. Дополнительные векторные и матричные функции

В этой главе рассматриваются внешние расширения системы Derive — как ориентированные под MS-DOS, так и под Windows. Эти расширения поставляются вместе с системой в виде библиотечных файлов вида \*.mth. В целом библиотеки Derive содержат определения сотен новых функций, большинство из которых едва ли понадобится конкретному пользователю. Однако эти функции наглядно иллюстрируют почти неисчерпаемые возможности расширения системы с помощью внешних библиотек. Пользователь может выбрать из множества библиотечных функций те, которые ему полезны, и составить свои библиотечные файлы. Библиотечные файлы Derive являются прекрасными примерами программирования на входном языке Derive многих математических задач — от простых до весьма специфических и сложных.

Библиотечные файлы системы отличаются от демонстрационных (см. раздел 3.10) тем, что в их наименовании стоит знак (C) защиты авторских прав фирмы Soft Warehous. Поэтому тексты таких файлов целиком в данной книге не приводятся. Пользователи могут ознакомиться с ними самостоятельно.

К сожалению, в фирменную поставку системы не входят демонстрационные примеры применения библиотечных файлов (их практически нет и в [21]). Поэтому полное перечисление всех определенных в этих файлах функций (с примерами применения многих из них) позволит читателям эффективно использовать дополнительные возможности системы.

Следует еще раз отметить, что для применения библиотечных функций необходимо загрузить файл с их определениями в память ПК. Для этого используются команды LOAD и UTILITY. Загрузка при опции UTILITY занимает несколько секунд (при опции LOAD загрузка занимает больше времени, так как текст файла выводится на экран дисплея).

В файле vector.mth задан ряд дополнительных векторных и матричных функций, использование которых базируется на применении команды SIMPLIFY. В оригинале руководства системы для этого употребляется термин “функция упрощается к ...”. Эти функции перечислены ниже (v и w — векторы, A и B — матрицы).

### Векторные операции (файл vector.mth):

|                              |                                                                                                         |
|------------------------------|---------------------------------------------------------------------------------------------------------|
| <b>i_</b>                    | — возвращает вектор [1, 0, 0];                                                                          |
| <b>j_</b>                    | — возвращает вектор [0, 1, 0];                                                                          |
| <b>k_</b>                    | — возвращает вектор [0, 0, 1];                                                                          |
| <b>OUTER (v, w)</b>          | — возвращает кронекерово (внешнее) произведение v и w;                                                  |
| <b>KRONECKER (i, j)</b>      | — возвращает дельта-функцию Кронекера, равную 1, если целое i равно целому j, и 0 — в противном случае; |
| <b>ADJOIN_ELEMENT (e, v)</b> | — присоединяет выражение e к началу вектора v;                                                          |

**APPEND\_VECTORS** ( $v, w$ )  
**DELETE\_ELEMENT** ( $v, k$ )

- объединяет вектор  $v$  с вектором  $w$ ;
- уничтожает  $k$ -й элемент вектора  $v$  и уменьшает размер вектора на 1;

**JACOBIAN** ( $u, v$ )

- возвращает Якобиан вектора  $x = u(c)$ , криволинейные координаты  $c$  и Декартовы  $x$ ;

**MINOR** ( $A, i, j$ )

- уничтожает строку  $i$  и столбец  $j$  матрицы  $A$  и уменьшает на 1 число строк и столбцов матрицы;

**REVERSE\_VECTOR** ( $v$ )

- возвращает вектор  $v$  с обратным порядком расположения элементов;

**SWAP\_ELEMENTS** ( $v, i, j$ )

- переставляет местами  $i$ -й и  $j$ -й элементы  $v$ ;

**SCALE\_ELEMENT** ( $v, i, s$ )

- перемножает  $i$ -й элемент  $v$  на  $s$ ;

**SUBTRACT\_ELEMENTS** ( $v, i, j$ ) — вычитает  $j$ -й элемент из  $i$ -го в векторе  $v$ .

Следующие примеры иллюстрируют применение некоторых из этих функций:

```

1: "Действие функций файла vector.mth"
2: j_
3: [0, 1, 0]
4: v := [0, 1, 2, 3, 4, 5]
5: OUTER (v, j_)
 / 0 0 0 \
 | 0 1 0 |
 | 0 2 0 |
6: | 0 3 0 |
 | 0 4 0 |
 \ 0 5 0 /
7: KRONECKER (2, 2)
8: 1
9: KRONECKER (0, 1)
10: 0
11: APPEND_VECTORS (v, v)
12: [0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 5]
13: v1 := DELETE_ELEMENT (v, 3)
14: v1
15: [0, 1, 3, 4, 5]
```

```

16: REVERSE_VECTOR (v)
17: [5, 4, 3, 2, 1, 0]
18: vs := SWAP_ELEMENTS (v, 2, 6)
19: [0, 5, 2, 3, 4, 1]
20: SCALE_ELEMENT (v, 3, 10)
21: [0, 1, 20, 3, 4, 5]
22: SUBTRACT_ELEMENTS (v, 2, 4)
23: [0, -2, 2, 3, 4, 5]

```

Кроме описанных выше векторных функций файл `vector.mth` содержит определения ряда дополнительных матричных функций.

**Матричные функции (файл `vector.mth`,  $v$  и  $w$  — векторы,  $A$  и  $B$  — матрицы):**

- ADJOINT (A)** — возвращает присоединенную или Эрмитово-сопряженную матрицу ( $A^{-1} \cdot \det(A)$ );
- COFACTOR (A, i, j)** — возвращает алгебраическое дополнение элемента  $(i, j)$  матрицы, обратной квадратной матрице  $A$ ;
- FORCE0 (A, i, j, p)** — обращает элемент  $(i, j)$  в 0 путем замены строки линейной комбинацией со строкой  $p$ ;
- GEOMETRY\_MATRIX (c, G)** — дает координатную геометрическую матрицу криволинейных координат  $c$  и метрического тензора  $G$ ;
- MATPROD (A, B, i, j)** — возвращает элемент  $(i, j)$  точечного произведения матриц  $A$  и  $B$ ;
- MINOR (A, i, j)** — удаляет  $i$ -ю строку и  $j$ -й столбец из  $A$ ;
- PIVOT (A, i, j)** — дает копию матрицы  $A$ , в которой столбец  $j$  под рядом  $i$  приведен к 0 вычитанием соответствующим образом умноженной строки  $i$ ;
- RANK (A)** — возвращает ранг матрицы  $A$  (число ненулевых рядов в матрице, приведенной к ступенчатой форме);
- SWAP\_ELEMENTS (v, i, j)** — переставляет  $i$ -й и  $j$ -й элементы вектора  $v$ ;
- SCALE\_ELEMENT (v, i, s)** — умножает  $i$ -й элемент вектора  $v$  на  $s$ ;
- APPROX\_EIGENVECTOR (A, mu)** — возвращает приближенный собственный вектор  $A$  (используйте команду `арргоX`);
- EXACT\_EIGENVECTOR (A, mu)** — возвращает собственный вектор  $A < 4 \times 4$  для собственного значения  $mu$ ;
- COVARIANT\_METRIC\_TENSOR (A)** — возвращает ковариантный метрический тензор Якобиевой матрицы  $A$ ;

**cylindrical**

— дает матрицу модулей базисных векторов для цилиндрических координат  $r$ ,  $\theta$ ,  $z$ ;

**spherical**

— дает координатную геометрическую матрицу для сферических координат  $r$ ,  $\theta$ ,  $\phi$ .

Следующие примеры показывают работу с некоторыми из этих матричных функций.

1: "Матричные функции файла vector.mth"

2: 
$$a := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 1 \end{pmatrix}$$

3: FORCE0 (a, 2, 2, 1)

4: 
$$\begin{pmatrix} 1 & 2 & 3 \\ 1.5 & 0 & -1.5 \\ 7 & 8 & 1 \end{pmatrix}$$

5: PIVOT (a, 2, 2)

6: 
$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0.6 & 0 & -8.6 \end{pmatrix}$$

7: 
$$b := \begin{pmatrix} 7 & 8 & 9 \\ 5 & 6 & 7 \\ 2 & 3 & 4 \end{pmatrix}$$

8: MATPROD (a, b, 1, 2)

9: 29

10: COFACTOR (a, 2, 2)

11: -20

12: EIGENVALUES (a, x)

13: [x = -5.07440]

14: EXACT\_EIGENVECTOR (a, x)

15: [ x1 = 0 x2 = 0 x3 = 0 ]

16: APPROX\_EIGENVECTOR (a, 1)

17: [0.726306, -0.683938, 0.0685955]

18: JACOBIAN ([1, 2, 3], [4, 5, 6])

19: 
$$\begin{array}{l} \left( \begin{array}{ccc} d & d & d \\ \frac{d}{d4} & \frac{d}{d5} & \frac{d}{d6} \\ d & d & d \\ \frac{d}{d4} & \frac{d}{d5} & \frac{d}{d6} \\ d & d & d \\ \frac{d}{d4} & \frac{d}{d5} & \frac{d}{d6} \end{array} \right) \end{array}$$

20: COVARIANT\_METRIC\_TENSOR (a)

21: 
$$\left( \begin{array}{ccc} 66 & 78 & 34 \\ 78 & 93 & 44 \\ 34 & 44 & 46 \end{array} \right)$$

22: GEOMETRY\_MATRIX (a, b)

23: 
$$\left( \begin{array}{ccc} [1, 2, 3] & [4, 5, 6] & [7, 8, 1] \\ 2.64575 & 2.44948 & 2 \end{array} \right)$$

24: cylindrical

25: 
$$\left( \begin{array}{ccc} r & \text{theta} & z \\ 1 & r & 1 \end{array} \right)$$

26: spherical

27: 
$$\left( \begin{array}{ccc} r & \text{theta} & \text{phi} \\ 1 & r \text{ SIN}(\text{phi}) & r \end{array} \right)$$

Действие большинства этих функций не вполне очевидно, поскольку большая часть из них относится к специальным разделам векторного и матричного анализа, включая тензорный анализ и теорию упругости. Их применение заметно расширяет возможности системы Derive в проведении специальных операций с векторами и матрицами [29]. К сожалению, характер и объем нашей книги не позволяют остановиться на более детальном описании указанных функций. Остается уповать, что те, кому это нужно, разберутся с более тонкими деталями применения описанных функций.

## 3.2. Решение нелинейных систем уравнений

Нелинейные системы уравнений (или системы нелинейных уравнений) редко имеют аналитические решения. Поэтому в систему Derive включены внешние расширения, позволяющие решать такие системы численными методами. Системы нелинейных уравнений могут иметь вид:

$$f_1(\dots x, y, z) = 0, \dots, f_n(\dots x, y, z) = 0.$$

Для решения таких уравнений известным методом Ньютона — Рафсона можно воспользоваться функцией [2]:

NEWTONS(u, x, x0, n)

Здесь u — вектор, содержащий функции  $f_1, \dots, f_n$ ; x — вектор переменных  $\dots x, y, z$  — аргументов функций; x0 — вектор начальных значений переменных  $\dots x, y, z$ ; n — число итераций. Функция при ее упрощении возвращает вектор решения. Если нужно найти решения, содержащие комплексные числа, надо использовать эти числа в начальном приближении.

В файле solve.mth функция NEWTONS определена так:

```
NEWTON_AUX(a, x, x0, n):=ITERATES(xk-
ELEMENT(ROW_REDUCE(LIM(a, x, xk)), DIMENSION(a)+1), xk, x0, n)
```

```
NEWTONS(u, x, x0, n):=NEWTON_AUX(APPEND(VECTOR(DIF (u,
ELEMENT(x, k_)), k_, 1, DIMENSION(x)), {u})', x, x0, n)
```

Обратите внимание на использование внутри задания функции NEWTONS внутренней переменной k\_. Знак “\_” используется, как уже отмечалось, для отличия этой переменной от переменной k, которая может задаваться без объявления.

Метод Ньютона — Рафсона является одним из лучших методов решения систем нелинейных уравнений. Как правило, он обеспечивает довольно быструю сходимость при решении систем нелинейных уравнений различного вида. Для использования внешней функции NEWTONS следует с помощью команд TRANSFER LOAD UTILITES загрузить файл solve.mth. Этот файл содержит определение еще одной функции для решения систем нелинейных уравнений итерационным методом:

```
FIXED_POINT(g, x, x0, n):=ITERATES(LIM(g, x, xk), xk, x0, n)
```

Эта функция используется в форме:

```
FIXED_POINT(g, x, x0, n)
```

Здесь g — вектор левых частей системы нелинейных уравнений, представленных в виде:

$$f_1(\dots x, y, z) = \dots, \dots, f_n(\dots x, y, z) = z.$$

В отличие от метода Ньютона — Рафсона, итерационный метод не всегда гарантирует сходимость решения. В результате решение может расходиться, вплоть до программной аварии системы. Оба реализованных в системе метода не предусматривают автоматической остановки при достижении погрешности решения заданного значения. Итерации выполняются за данное число раз. Анализ возвращаемой матрицы решений позволяет судить об эффективности и завершенности решения.

В качестве примера рассмотрим решение нелинейной системы из двух уравнений методом Ньютона — Рафсона:

$$\begin{aligned}x^2 + y^2 - 5 &= 0, \\x + y - 3 &= 0.\end{aligned}$$

Диалог с Derive для этого примера имеет следующий вид:

```

1: "Решение системы нелинейных уравнений"
2: NEWTONS ($\langle x^2 + y^2 - 5, x + y - 3 \rangle$, [x, y], [0.8, 2.2], 5)
3:
 / 0.8 2.2 \
 | 0.971428 2.02857 |
 | 0.999227 2.00077 |
 | 1 2 |
 | 1 2 |
 | 1 2 |
 \ 1 2 /
4: FIXED_POINT (SIN (x) + 0.25, x, 1.2, 5)
5: [1.2, 1.18203, 1.17538, 1.17283, 1.17185, 1.17147]

```

Как видно из этого примера, функция NEWTONS возвращает вектор переменных  $x$  и  $y$  для заданных  $n$ -итераций. Видно, что, начиная уже с третьей итерации, решение перестает изменяться, так что найденные значения переменных следующие:  $x = 1$  и  $y = 2$ .

В строках 4 и 5 с помощью функции FIX\_POINT решается нелинейное уравнение:

$$\sin(x) + 0.25 * x = x.$$

Из строки 5 видно, что даже для такого простого уравнения сходимость к решению более медленная, чем при реализации метода Ньютона — Рафсона. Ситуация при решении систем нелинейных уравнений может быть еще хуже. Тем не менее для решения систем определенного вида итерационный метод может иметь преимущества.

Следует помнить, что не всегда решения нелинейных уравнений возможны. Если решение не удастся получить при 9–10 итерациях, нет смысла увеличивать их число, разумнее подыскать более подходящие начальные приближения. Если решение идет явно долго и не приводит к выводу результатов, это, скорее всего, означает расходимость решения. При этом промежуточные результаты могут оказаться чрезмерно громоздкими и вызвать крах решения с сообщением о нехватке памяти. Тут разумно остановить вычисления нажатием клавиши Esc. Попробуйте также для подобного случая использовать команду arpoX, если вы ранее применяли Simplify. Помните также, что решения в виде матриц помещаются на экран, если  $n$  меньше 10.

### 3.3. Решение дифференциальных уравнений

Решение многих задач в математике, физике и других областях науки и техники связано с решением дифференциальных уравнений и систем из них. В частности, такие уравнения обычно описывают поведение динамических



систем во времени. Решениями дифференциальных уравнений могут быть специальные математические функции. Derive содержит три библиотечных файла внешних расширений, ориентированных на решение дифференциальных уравнений различного типа.

### 3.3.1. Частное решение для начальных условий $y = y_0$ при $x = x_0$

В файле ode1.mth содержатся функции для решения различных дифференциальных уравнений первого порядка с применением команды Simplify. Ниже перечислены эти функции.

**Функция файла ode1.mth**

**DSOLVE1** (p, q, x, y, x0, y0)  
**SEPARABLE** (p, q, x, y, x0, y0)  
**LINEAR1\_GEN** (p, q, x, y, c)

**LINEAR1** (p, q, x, y, x0, y0)  
**HOMOGENEOUS\_GEN** (r, x, y, c)  
**HOMOGENEOUS** (r, x, y, x0, y0)

**EXACT\_GEN** (p, q, x, y, c)

**EXACT** (p, q, x, y, x0, y0)

**INTEGRATING\_FACTOR**  
 (p, q, x, y, x0, y0)

**Частное решение уравнения**

$p(x, y) + q(x, y)y' = 0$  — частное решение  
 $y' = p(x)q(y)$  — частное решение  
 $y' + p(x)y = q(x)$  — общее решение, содержащее константу c  
 $y' + p(x)y = q(x)$  — частное решение  
 $y' = r(x, y)$  — общее решение (r однородно)  
 $y' = r(x, y)$  — частное решение (r однородно)  
 $p(x, y) + q(x, y)y' = 0$  — общее решение, если оно в полных дифференциалах  
 $p(x, y) + q(x, y)y' = 0$  — частное решение, если оно в полных дифференциалах  
 $p(x, y) + q(x, y)y' = 0$  — если уравнение становится уравнением в полных дифференциалах при умножении на интегрирующий множитель, не содержащий x или y

Если решение не удовлетворяет отмеченным условиям, то возвращается сообщение “inapplicable” (“неприменимо”).

### 3.3.2. Общее решение, содержащее константу c

Для получения общего решения, содержащего константу c, в файле ode1.mth находятся следующие функции:

**Функция файла ode1.mth**

**DSOLVE1\_GEN**(p, q, x, y, c)  
**SEPARABLE\_GEN** (p, q, x, y, c)  
**LINEAR1\_GEN** (p, q, x, y, c)  
**HOMOGENEOUS\_GEN** (r, x, y, c)  
**EXACT\_GEN** (p, q, x, y, c)

**INTEGRATING\_FACTOR\_GEN**  
 (p, q, x, y, c)

**Общее решение уравнения**

$p(x, y) + q(x, y)y' = 0$   
 $y' = p(x)q(y)$   
 $y' + p(x)y = q(x)$   
 $y' = r(x, y)$ , если r однородно  
 $p(x, y) + q(x, y)y' = 0$ , если оно в полных дифференциалах  
 $p(x, y) + q(x, y)y' = 0$ , если уравнение становится уравнением в полных дифференциалах при умножении на интегрирующий множитель, не содержащий x или y

Если решение не удовлетворяет отмеченным условиям, то возвращается сообщение “inapplicable” (“неприменимо”).

### 3.3.3. Обыкновенные дифференциальные уравнения первого порядка (развитые методы)

Следующие функции файла ode1.mth поддерживают развитые методы решения обыкновенных дифференциальных уравнений первого порядка:

| Функция файла ode1.mth                          | Решаемое уравнение                                                                               |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------|
| MONOMIAL_TEST (p, q, x, y)                      | Интегрирующий множитель уравнения $p(x, y) + q(x, y)y' = 0$ , если результат имеет вид $x^m y^n$ |
| BERNOULLI (p, q, k, x, y, x0, y0)               | $y' + p(x)y = q(x)y^k$                                                                           |
| GEN_HOM (r, x, y, x0, y0)                       | $y' = r(x, y)$ , если r обобщенное однородное                                                    |
| FUN_LIN_CCF<br>(r, p, q, k, x, y, x0, y0)       | $y' = r(p * x + q * y + k)$ , если p, q, k — константы                                           |
| LIN_FRAC<br>(r, a, b, c, p, q, k, x, y, x0, y0) | $y' = r((ax + by + c)/(px + qy + k))$                                                            |
| ALMOST_LIN<br>(r, b, p, q, x, y, x0, y0)        | $r(x, y)y' + p(x)b(y) = q(x)$ , если оно квазилинейно, т. е. $(db/dy)/r$ не содержит y           |

Для этих уравнений общее решение, содержащее константу c, выполняется с помощью приведенных ниже функций.

| Функция файла ode1.mth                         | Решаемое уравнение                                     |
|------------------------------------------------|--------------------------------------------------------|
| BERNOULLI_GEN (p, q, k, x, y, c)               | $y' + p(x)y = q(x)y^k$                                 |
| GEN_HOM_GEN (r, x, y, c)                       | $y' = r(x, y)$ , если r обобщенное однородное          |
| FUN_LIN_CCF_GEN (r, p, q, k, x, y, c)          | $y' = r(p * x + q * y + k)$ , если p, q, k — константы |
| LIN_FRAC_GEN<br>(r, a, b, c, p, q, k, x, y, c) | $y' = r((ax + by + c)/(px + qy + k))$                  |
| ALMOST_LIN_GEN (r, b, p, q, x, y, c)           | $r(x, y)y' + p(x)b(y) = q(x)$ , если оно квазилинейно  |
| CLAIRAUT (p, q, x, y, v, c)                    | $p(x*v - y) = q(v)$ , где v обозначает y'              |

Большинство приведенных функций относится к решению дифференциальных уравнений первого порядка специальных видов. Функции возвращают матрицу решений. Функции, содержащие суффикс GEN, возвращают общее решение, имеющее символическую константу. Функции без этого суффикса возвращают частное решение, если заданы численные начальные условия, и общее решение, содержащее начальные условия в символическом виде, если начальные условия были заданы символическими переменными. Если решение не удовлетворяет отмеченным условиям, то возвращается сообщение “inapplicable” (“неприменимо”).

Результат обычно является алгебраическим выражением, неявно определяющим независимую переменную от зависимой. Он может содержать интегралы, которые Derive не может “взять”. Такие решения в математике являются корректными. Если неявное выражение не содержит интегралов, можно использовать функцию SOLVE для поиска одного или нескольких явных решений.

### 3.3.4. Примеры решения дифференциальных уравнений первого порядка

Приведем примеры решения дифференциальных уравнений первого порядка.

1: “Решение дифференциальных уравнений первого порядка”

2: LINEAR1 (p, q, x, y, x0, y0)

$$3: y = \frac{q}{p} - \frac{q \#e^{-p x}}{p}$$

4: LINEAR1 (2 x, x<sup>2</sup>, x, y, 0, 0)

$$5: y = \#e^{-\int(2x, x, 0, x)} \left/ \int_0^x 2 \int(2x, x, 0, x) dx \right. \#e^{\int(2x, x, 0, x)}$$

$$6: y = \frac{x}{2} + \frac{\#i \text{SQRT}(pi) \#e^{-x} \text{ERF}(\#i x)}{4}$$

7: BERNOULLI (2 x, x<sup>2</sup>, 0, x, y, 0, 0)

$$8: y = \#e^{-\int(2x, x, 0, x)} \left/ \int_0^x 2 \int(2x, x, 0, x) dx \right. \#e^{\int(2x, x, 0, x)}$$

9: FUN\_LIN\_CCF (2 x, 1, 1, 1, x, y, 0, 0)

$$10: \int_1^{x+y+1} \frac{1}{2x+1} dy - \int_0^x 1 dx = 0$$

$$11: \frac{2y-1}{2(2x+1)} - x + \frac{1}{2} = 0$$

12: CLAIRAUT (x<sup>2</sup>, x, x, y, 1, 1)

$$13: \left\{ \lim_{x \rightarrow 1} (x^2 - x) = 0, -\frac{d}{dx} x = 0 \right\}$$

### 3.3.5. Обыкновенные дифференциальные уравнения второго порядка

Файл ode2.mth содержит ряд функций (включая тестовые со словом TEST в конце имени, проверяющие допустимость применения основных функций) для решения дифференциальных уравнений второго порядка.

Решение дифференциальных уравнений второго порядка обеспечивается следующими функциями файла ode2.mth:

| Функция файла ode2.mth                           | Решаемое уравнение                                                                              |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------|
| DSOLVE2 (p, q, r, x, c1, c2)                     | $y'' + p(x)y' + q(x)y = r(x)$                                                                   |
| DSOLVE2_BV<br>(p, q, r, x, x0, y0, x2, y2)       | $y'' + p(x)y' + q(x)y = r(x)$                                                                   |
| DSOLVE2_IV (p, q, r, x, x0, y0, v0)              | $y'' + p(x)y' + q(x)y = r(x)$                                                                   |
| AUTONOMOUS_CONSERVATIVE<br>(q, x, y, x0, y0, v0) | $y'' = q(y)$ , $y = y_0$ и $y' = v_0$ при $x = x_0$                                             |
| LIOUVILLE (p, q, x, y, c1, c2)                   | $y'' + p(x)y' + q(y)(y')^2 = 0$                                                                 |
| AUTONOMOUS (r, v) := dv/dy                       | заданное уравнение $y'' = r(y, v)$ приводит к двум последовательным уравнениям 1-го порядка     |
| EXACT2 (p, q, x, y, v, c)                        | понижает порядок $p(x, y, v)y'' + q(x, y, v) = 0$ с $v = y'$ , если оно в полных дифференциалах |
| LIN2_POS (p, q, r, x)                            | $y'' + p(x)y' + q(x)y = r(x)$ , если LIN2_TEST (p, q, x) не зависит от x и положительна         |
| LIN2_NEG (p, q, r, x)                            | $y'' + p(x)y' + q(x)y = r(x)$ , если LIN2_TEST (p, q, x) не зависит от x и отрицательна         |
| LIN2_0 (p, q, r, x)                              | $y'' + p(x)y' + q(x)y = r(x)$ , если LIN2_TEST (p, q, x) есть 0                                 |

Если решение невозможно, возвращается сообщение "inapplicable" ("неприменимо"). Работу некоторых из этих функций поясняют следующие примеры:

1: "Решение дифференциальных уравнений второго порядка"

2: LIN2\_POS (a, b, 0, x)

$$3: c1 \#e \quad x \left( \sqrt{a^2 - 4b} / 2 - a/2 \right) - \frac{c2 \#e - x \left( \sqrt{a^2 - 4b} + a \right) / 2}{\sqrt{a^2 - 4b}}$$

4: LIN2A\_POS (0, a, 0, x)

$$5: c1 \#e \quad \frac{\#1 \sqrt{a} x}{2 \sqrt{a}} + \frac{- \#1 \sqrt{a} x}{2 \sqrt{a}}$$

```
6: LIOUVILLE (0, 1, x, y, 1, 1)
```

```
7: #ey + x - 1 = 0
```

### 3.3.6. Решение уравнения вида $y' = r(x)$ с начальным условием $y(x_0) = y_0$

Файл `ode_appr.mth` содержит дополнительные расширения для решения дифференциальных уравнений. Для решения одиночных дифференциальных уравнений вида  $y' = r(x)$  при  $y(x_0) = y_0$  в нем заданы функции, представленные ниже.

- DIRECTION\_FIELD** ( $r, x, x_0, x_m, m, y, y_0, y_n, n$ ) — строит сетку с координатами  $(x_0, y_0)$  и  $(x_m, y_n)$  противоположащих углов (используются команды `apprX` и `Plot`);
- TAY\_ODE1** ( $r, x, y, x_0, y_0, n$ ) — решение в виде многочлена Тейлора  $n$ -й степени;
- PICARD** ( $r, p, x, y, x_0, y_0$ ) — уточнение приближения по заданному приближению  $p(x)$ ;
- EULER**( $r, x, y, x_0, y_0, h, n$ ) — дает матрицу  $[[x_0, y_0], \dots, [x_n, y_n]]$  решения методом Эйлера при шаге  $h$  (используется команда `apprX`).

На основе применения этих функций в файле `ode_appr.mth` определены также и функции для решения систем обыкновенных дифференциальных уравнений, заданных в виде векторов  $y' = r(x)$  и вектора начальных условий  $y_0 = y(x_0)$ :

- TAY\_ODES** ( $r, x, y, x_0, y_0, n$ ) — дает решение в виде рядов Тейлора  $n$ -й степени;
- PICARD**( $r, p, x, y, x_0, y_0$ ) — дает улучшение аппроксимации в виде вектора  $p(x)$ ;
- RK** ( $r, v, v_0, h, n$ ) — дает решение методом Рунге — Кутта в виде матрицы  $[[x_0, y_0(x_0), y_m(x_0)], [x_n, y_0(x_n), y_m(x_n)]]$  с шагом решения  $h$  (используйте команду `apprX`);
- EXTRACT\_2\_COLUMNS** ( $A, j, k$ ) — выделяет столбцы  $j, k$  матрицы  $A$  для графического представления результатов решения методом Рунге — Кутта.

### 3.3.7. Функция Рунге — Кутта

Из этих функций файла `ode_appr.mth` наиболее важной является функция `RK`, обеспечивающая решение систем нелинейных дифференциальных уравнений численным методом Рунге — Кутта [2]. В указанном файле эта функция определена следующим образом:

```
RK_AUX3(p, v, u_, c1, c2, c3): = (c1 + LIM(p, v, u_ + c3) + 2 * (c2 + c3))/6
```

```
RK_AUX2(p, v, u_, c1, c2): = RK_AUX3(p, v, u_, c1, c2, LIM(p, v, u_ + c2/2))
```

```
RK_AUX1(p, v, u_, c1): = RK_AUX2(p, v, u_, c1, LIM(p, v, u_ + c1/2))
```

$$\text{RK\_AUX0}(p, v, v0, n) := \text{ITERATES}(u\_ + \text{RK\_AUX1}(p, v, u\_ , \text{LIM}(p, v, u\_ )), u\_ , v0, n)$$

$$\text{RK}(r, v, v0, h, n) := \text{RK\_AUX0}(h * \text{APPEND}(\{1\}, r), v, v0, n)$$

Это определение дает прекрасный пример техники программирования на языке системы *Derive* и свидетельствует о компактности программы. К примеру, программы решения системы дифференциальных уравнений методом Рунге — Кутты на Бейсике или Паскале по крайней мере в несколько раз длиннее, чем приведенная программа из пяти строк. Она является наглядным примером функционального программирования, когда конечная функция определена через ряд промежуточных функций.

Функция *RK* в документах используется в виде:

**RK(r, v, v0, h, n).**

В этой функции *r* — вектор левых частей системы обыкновенных дифференциальных уравнений первого порядка:

$$y' = f_1(\dots, x, y, z) \dots z' = f_n(\dots, x, y, z).$$

Вектор *v* должен содержать перечисление всех переменных *...x, y, z*, а вектор *v0* — их начальные значения. Шаг решения по переменной *x* задается значением *h*, а число точек решения — значением *n*. Решение представляется в виде матрицы, столбцы которой дают значения переменных *...x, y, z*.

Рассмотрим решение следующей системы дифференциальных уравнений:

$$y' = z \quad \text{и} \quad z' = (y/x - z)/x - y$$

с начальными условиями:  $x = 0.3, y = 0.1484$  и  $z = 0.483$ . При  $h = 0.1$  и  $n = 7$  решение представлено ниже.

1: "Решение системы дифференциальных уравнений"

2: 
$$\text{RK} \left( \begin{array}{l} // \quad y \quad \backslash \\ || \quad \text{---} \text{---} z \quad | \\ || \quad x \quad | \\ || z, \text{---} \text{---} y \quad | \\ \backslash \quad x \quad / \end{array} \right), [x, y, z], [0.3, 0.1484, 0.483], 0.1, 7$$

3: 
$$\begin{array}{l} / \quad 0.3 \quad 0.1484 \quad 0.483 \quad \backslash \\ | \quad 0.4 \quad 0.196090 \quad 0.470204 \quad | \\ | \quad 0.5 \quad 0.242322 \quad 0.453852 \quad | \\ | \quad 0.6 \quad 0.286748 \quad 0.434113 \quad | \\ | \quad 0.7 \quad 0.329038 \quad 0.411164 \quad | \\ | \quad 0.8 \quad 0.368881 \quad 0.385200 \quad | \\ | \quad 0.9 \quad 0.405985 \quad 0.356439 \quad | \\ \backslash \quad 1 \quad 0.440083 \quad 0.325121 \quad / \end{array}$$

Несмотря на принципиальную возможность решения систем дифференциальных уравнений средствами *Derive*, следует все же отметить, что по

скорости счета такие решения заметно проигрывают специализированным программам, реализованным на языках высокого уровня, например Бейсике [2], Форте [3] или Паскале. Это естественная плата за ориентацию языка реализации системы Derive — LISP на решение задач в символьном, а не в численном виде.

### 3.4. Решение рекуррентных уравнений

Многие уравнения имеют простые рекуррентные решения, т. е. такие решения, когда на каждом очередном шаге новое решение находится по предшествующему решению. Рекуррентные уравнения могут решаться намного быстрее, чем, например, дифференциальные уравнения обычными методами (такими, как метод Рунге — Кутты), поскольку описывающие их рекуррентные формулы нередко очень просты.

Решение рекуррентных уравнений задано в файле `recureqn.mth` рядом функций:

|                                                  |                                                                       |
|--------------------------------------------------|-----------------------------------------------------------------------|
| <b>LIN1_DIFFERENCE</b> (p, q, x, x0, y0)         | — решение уравнений $y(x+1) = p(x)y(x) + q(x)$ при $y(x_0) = y_0$ ;   |
| <b>RECURRENCE1</b> (r, x, y, x0, y0, n)          | — n шагов решения $y(x+1) = r(x), y(x_0) = y_0$ ;                     |
| <b>GEOMETRIC1</b> (k, p, q, x, x0, y0)           | — решение $y(k * x) = p(x)y(x) + q(x)$ при $y(x_0) = y_0$ ;           |
| <b>CLAIRAUT_DIF</b> (p, q, d, x, y)              | — решение $p(y - xd) = q(d)$ , где d есть $y(x+1) - y(x)$ ;           |
| <b>LIN2_CCF_DISC</b> (p, q)                      | — дискриминант d для $y(x+2) + p * y(x+1) + q * y(x) = r(x)$ ;        |
| <b>LIN2_CCF_POS</b> (p, q, r, x)                 | — решение $y(x+2) + p * y(x+1) + q * y(x) = r(x)$ для $d > 0$ ;       |
| <b>LIN2_CCF_NEG</b> (p, q, r, x)                 | — решение $y(x+2) + p * y(x+1) + q * y(x) = r(x)$ для $d < 0$ ;       |
| <b>LIN2_CCF_0</b> (p, q, r, x)                   | — решение $y(x+2) + p * y(x+1) + q * y(x) = r(x)$ для $d = 0$ ;       |
| <b>IMPOSE_BV2</b> (x, y, x0, y0, x2, y2, c1, c2) | — дает c1 и c2 для $y(x)$ , так что $y(x_0) = y_0$ и $y(x_2) = y_2$ . |

Ниже представлены примеры решения рекуррентных уравнений.

1: "Решение рекуррентных уравнений"

2: LIN1\_DIFFERENCE (x<sup>2</sup>, x, x, a, b)

$$3: \frac{(x-1)! \sum_{i=1}^x \frac{a^{i-1}}{(i-1)!} (a-1)^{x-i} + b}{(a-1)!^2}$$

4: CLAIRAUT\_DIF (x<sup>2</sup>, x, d, x, y)

```

5: x2 - x = 0
6: LIN2_CCF_DISC (x2, x)
7: x4 - 4 x
8: LIN2_CCF_POS (x2, x, 1, 0)
9: c1 + c2

```

Как видно из приведенного примера, решения могут быть найдены в символьном виде. При этом используются команды основного меню Simplify и approx.

## 3.5. Дополнительные функции дифференцирования

В файле numeric.mth содержится ряд дополнительных функций вычисления производных и сглаживания численными методами. Для вычисления производных используются простые формулы численного дифференцирования, что заметно ускоряет выполнение вычислений (применяется команда approx). Ниже приведены функции файла numeric.mth с указанием возвращаемого ими значения.

### 3.5.1. Функции численного дифференцирования

Для численного дифференцирования в файле numeric.mth содержатся следующие функции:

**DIF\_NUM** (y, x, x0, h) — дает первую частную производную y по x в точке x0 при шаге изменения x — h;

**DIF2\_NUM** (y, x, x0, h) — дает вторую частную производную y по x;

**SMOOTH\_VECTOR** (v) — дает сглаженную копию вектора v (каждый элемент есть среднее между ним и двумя соседними элементами, крайние элементы сохраняются);

**SMOOTH\_COLUMN** (A, j) — дает матрицу A со сглаженным столбцом j;

**DIF\_DATA** (A) — дает сглаженную первую производную для двухстолбцовой матрицы данных A;

**DIF2\_DATA** (A) — дает сглаженную вторую производную матрицы A;

**INT\_DATA** (A) — дает сглаженную производную матрицы A с применением правила трапеций.

Ниже представлено решение задач с перечисленными функциями.

```

1: "Численное дифференцирование и сглаживание"
2: DIF_NUM (SIN (x), x, 1, 0.001)
3: 0.540290
4: DIF2_NUM (SIN (x), x, 1, 0.001)
5: -0.866002

```



```
6: SMOOTH_VECTOR [0, 1.1, 2.1, 2.8, 4.1, 5.2]
```

```
7: [0, 1.06666, 2, 3, 4.03333, 5.2]
```

```
8: SMOOTH_COLUMN $\left(\begin{array}{c} // \quad 0 \quad \backslash \\ | \quad 1.1 \quad | \\ | \quad 2.1 \quad | \\ | \quad 2.8 \quad | \\ | \quad 4.1 \quad | \\ \backslash \quad 5.2 \quad / \end{array} \right), 1$
```

```
9: $\left(\begin{array}{c} / \quad 0 \quad \backslash \\ | \quad 1.06666 \quad | \\ | \quad 2 \quad | \\ | \quad 3 \quad | \\ | \quad 4.03333 \quad | \\ \backslash \quad 5.2 \quad / \end{array} \right)$
```

```
10: INT_DATA $\left(\begin{array}{c} / \quad 1 \quad 1 \quad \backslash \\ | \quad 2 \quad 2 \quad | \\ \backslash \quad 3 \quad 3 \quad / \end{array} \right)$
```

```
11: $\left(\begin{array}{c} / \quad 1 \quad 0 \quad \backslash \\ | \quad 2 \quad 1 \quad | \\ \backslash \quad 3 \quad 2 \quad / \end{array} \right)$
```

В другом файле dif\_apps.mth содержится еще ряд функций, базирующихся на операции дифференцирования — см. ниже.

### 3.5.2. Применение операций дифференцирования

Применение операций дифференцирования реализуется следующими функциями файла dif\_apps.mth:

**CURVATURE** (y, x)

**CENTER\_OF\_CURVATURE** (y, x)

**TANGENT** (y, x, x0)

**PERPENDICULAR** (y, x, x0)

**OSCULATING\_CIRCLE** (y, x, theta)

- кривизна зависимости  $y(x)$ ;
- вектор центра кривизны линии  $y(x)$ ;
- касательная к линии  $y(x)$  в точке  $x = x_0$ ;
- перпендикуляр к линии  $y(x)$  в точке  $x = x_0$ ;
- соприкасающаяся с линией  $y(x)$  окружность, заданная параметрически с параметром  $\theta$ ;

|                                                           |                                                                                                             |
|-----------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>PARA_DIF</b> ( $v, t, n$ )                             | — $n$ -я производная $v$ по $t$ , где $v = [x(t), y(t)]$ ;                                                  |
| <b>PARA_CURVATURE</b> ( $v, t$ )                          | — кривизна параметрической кривой $v$ ;                                                                     |
| <b>PARA_CENTER_OF_CURVATURE</b> ( $v, t$ )                | — вектор центра кривизны $v$ ;                                                                              |
| <b>PARA_TANGENT</b> ( $v, t, t_0, x$ )                    | — касательная к $v$ при $t = t_0$ , выраженная через $x$ ;                                                  |
| <b>PARA_PERPENDICULAR</b> ( $v, t, t_0, x$ )              | — перпендикуляр к $v$ при $t = t_0$ ;                                                                       |
| <b>PARA_OSCULATING_CIRCLE</b> ( $v, t, t_0, phi$ )        | — соприкасающаяся при $t = t_0$ с $v$ окружность, заданная параметрически с параметром $phi$ ;              |
| <b>POLAR_DIF</b> ( $r, theta, n$ )                        | — $n$ -я производная $r(theta)$ в функции от $theta$ ;                                                      |
| <b>POLAR_CURVATURE</b> ( $r, theta$ )                     | — кривизна функции $r(theta)$ в полярной системе координат;                                                 |
| <b>POLAR_CENTER_OF_CURVATURE</b> ( $r, theta$ )           | — центр кривизны функции в полярной системе координат;                                                      |
| <b>POLAR_TANGENT</b> ( $r, theta, theta_0, x$ )           | — касательная к $r(theta)$ при $theta = theta_0$ в функции от $x$ ;                                         |
| <b>POLAR_PERPENDICULAR</b> ( $r, theta, theta_0, x$ )     | — перпендикуляр к радиусу окружности при $theta = theta_0$ ;                                                |
| <b>IMP_DIF</b> ( $u, x, y, n$ )                           | — неявная производная $DIF(u, x, y)$ для $u(x, y) = 0$ при $n$ целом положительном (по умолчанию равным 1); |
| <b>IMP_CURVATURE</b> ( $u, x, y$ )                        | — кривизна для неявной кривой $u = 0$ ;                                                                     |
| <b>IMP_CENTER_OF_CURVATURE</b> ( $u, x, y$ )              | — центр для кривизны $u = 0$ ;                                                                              |
| <b>IMP_TANGENT</b> ( $u, x, y, x_0, y_0$ )                | — касательная линия к $u = 0$ при $x = x_0$ и $y = y_0$                                                     |
| <b>IMP_PERPENDICULAR</b> ( $u, x, y, x_0, y_0$ )          | — перпендикуляр $u = 0$ при заданных $x_0$ и $y_0$ ;                                                        |
| <b>IMP_OSCULATING_CIRCLE</b> ( $u, x, y, x_0, y_0, phi$ ) | — соприкасающаяся в точке $(x_0, y_0)$ окружность, заданная параметрически с параметром $phi$ ;             |
| <b>TANGENT_PLANE</b> ( $u, v, v_0$ )                      | — плоскость, касательная к $u(x, y, z) = 0$ при $[x, y, z] = v = v_0$ ;                                     |
| <b>NORMAL_LINE</b> ( $u, v, v_0, t$ )                     | — декартовский координатный вектор, линейный по переменной $t$ (нормаль к $u = 0$ при $v = v_0$ ).          |

Применение функций дифференцирования поясняют следующие примеры:

1: "Применение некоторых функций дифференцирования"

2: CURVATURE  $(x, x)$

```

3:
$$\frac{2}{(4x^2 + 1)^{3/2}}$$

4: CENTER_OF_CURVATURE (x2, x)
5:
$$\left(-4x^3, 0.5(6x^2 + 1) \right)$$

6: TANGENT (x2, x, 1)
7: 2x - 1
8: PERPENDICULAR (x2, x, 1)
9: -0.5(x - 3)
10: OSCULATING_CIRCLE (x2, x, 0)
11:
$$\left(0.5(4x^2 + 1)^{1.5} - 4x^3, 0.5(6x^2 + 1) \right)$$

12: POLAR_DIF (r, theta, 2)
13:
$$-\frac{\cos^2(\theta)}{r \sin^3(\theta)} - \frac{1}{r \sin(\theta)}$$

14: NORMAL_LINE (u, v, v 0, t)
15: t GRAD (u, 0)

```

Большинство функций файла `dif_apps` служит для решения специальных геометрических задач. Ввиду их очевидности тестирование задач, не попавших в приведенные примеры, может быть выполнено заинтересованным пользователем самостоятельно.

### 3.6. Дополнительные функции интегрирования

Файл `int_apps.mth` содержит большое количество функций, базирующихся на численном интегрировании или представляющих интегральные преобразования специального вида. Здесь также функции для вычисления длин дуг, площадей различных фигур и центров масс фигур, заполненных веществом с заданной функцией распределения плотности  $\mu$ . Наибольший интерес представляет вычисление  $n$  гармоник ряда Фурье для произвольной функции

$y(x)$  и преобразование Лапласа. Обе эти функции широко используются в практике электротехнических и радиотехнических расчетов.

Кроме того, в этом файле представлены функции для расчета по интегральным выражениям площади (в названии — слово AREA) и объема (в названии — слово VOLUME) для ряда фигур в выделенной (ограниченной) их области. Через параметры  $x, y, z$  задаются аналитические выражения для описания фигур, а через параметры  $x_1, y_1, z_1$  и  $x_2, y_2, z_2$  указываются области ограничения фигур. Ниже перечислены функции файла `int_apps.mth` с указанием возвращаемого ими значения.

#### Применение методов интегрирования (файл `int_apps.mth`):

- FOURIER** ( $y, t, t_1, t_2, n$ ) — серия из  $n$  гармоник тригонометрического ряда Фурье, приближающего функцию  $y(t)$  в интервале  $t$  от  $t_1$  до  $t_2$ ;
- LAPLACE** ( $y, t, s$ ) — интегральное преобразование Лапласа для функции  $y(t)$ , дающее функцию от независимой переменной  $s$ ;
- ARC\_LENGTH** ( $y, x, x_1, x_2$ ) — длина дуги функции  $y(x)$  для  $x$  в интервале от  $x_1$  до  $x_2$ ;
- ARC\_LENGTH** ( $y, x, x_1, x_2, \mu$ ) — интеграл  $\mu(x)$  по всей линии дуги  $\arcsin y(x)$  для  $x$  от  $x = x_1$  до  $x_2$ ;
- POLAR\_ARC\_LENGTH** ( $r, \theta, \theta_1, \theta_2$ ) — длина полярной дуги  $r(\theta)$  при изменении  $\theta$  от  $\theta_1$  до  $\theta_2$ ;
- POLAR\_ARC\_LENGTH** ( $r, \theta, \theta_1, \theta_2, \mu$ ) — интеграл  $\mu(\theta)$  по всей линии дуги  $r$ ;
- PARA\_ARC\_LENGTH** ( $v, t, t_1, t_2$ ) — длина дуги для вектора  $v(t)$  при  $t$  от  $t = t_1$  до  $t_2$ ;
- PARA\_ARC\_LENGTH** ( $v, t, t_1, t_2, \mu$ ) — интеграл  $\mu(t)$  по всей линии  $v$  при  $t$  от  $t = t_1$  до  $t_2$ ;
- AREA** ( $x, x_1, x_2, y, y_1, y_2, \mu$ ) — интеграл  $\mu(x, y)$  по области  $x$  от  $x_1$  до  $x_2$  и для  $y$  от  $y_1(x)$  до  $y_2(x)$ ;
- AREA** ( $x, x_1, x_2, y, y_1, y_2$ ) — площадь области при изменении  $x$  от  $x_1$  до  $x_2$  и  $y$  от  $y_1(x)$  до  $y_2(x)$ ;
- AREA\_CENTROID** ( $x, x_1, x_2, y, y_1, y_2$ ) — геометрический центр масс области при изменении  $x$  от  $x_1$  до  $x_2$  и  $y$  от  $y_1(x)$  до  $y_2(x)$ ;
- AREA\_CENTROID** ( $x, x_1, x_2, y, y_1, y_2, \mu$ ) — центр масс ограниченной области с распределением плотности  $\mu(x, y)$ ;
- AREA\_INERTIA** ( $x, x_1, x_2, y, y_1, y_2$ ) — геометрический тензор инерции для ограниченной области;
- AREA\_INERTIA** ( $x, x_1, x_2, y, y_1, y_2, \mu$ ) — тензор инерции для ограниченной области с распределением плотности  $\mu(x, y)$ ;

- POLAR\_AREA** ( $r, r1, r2, th, th1, th2$ ) — площадь области в полярной системе координат при  $th$ , меняющемся от  $th1$  до  $th2$  и  $r$  от  $r1(th)$  до  $r2(th)$ , при этом  $th$  — угол,  $r$  — длина радиус-вектора;
- POLAR\_AREA** ( $r, r1, r2, th, th1, th2, mu$ ) — интеграл  $mu(th)$  по ограниченной области в полярной системе координат;
- SURFACE\_AREA** ( $z, x, x1, x2, y, y1, y2$ ) — площадь поверхности  $z(x, y)$  с заданными ограничениями  $x$  от  $x1$  до  $x2$  и  $y$  от  $y1(x)$  до  $y2(x)$ ;
- SURFACE\_AREA** ( $z, x, x1, x2, y, y1, y2, mu$ ) — интеграл  $mu(x, y)$  по ограниченной поверхности  $z(x, y)$ ;
- VOLUME** ( $x, x1, x2, y, y1, y2, z, z1, z2$ ) — объем области при изменении  $x$  от  $x1$  до  $x2$ ,  $y$  от  $y1(x)$  до  $y2(x)$  и  $z$  от  $z1(x, y)$  и  $z2(x, y)$ ;
- VOLUME** ( $x, x1, x2, y, y1, y2, z, z1, z2, mu$ ) — интеграл  $mu(x, y, z)$  по ограниченной области при изменении  $x$  от  $x1$  до  $x2$ ,  $y$  от  $y1(x)$  до  $y2(x)$  и  $z$  от  $z1(x, y)$  и  $z2(x, y)$ ;
- VOLUME\_CENTROID** ( $x, x1, x2, y, y1, y2, z, z1, z2$ ) — объемный тензор инерции для ограниченной области (см. выше);
- VOLUME\_CENTROID** ( $x, x1, x2, y, y1, y2, z, z1, z2, mu$ ) — центр масс для ограниченной области с распределением плотности  $mu(x, y, z)$ ;
- VOLUME\_INERTIA** ( $x, x1, x2, y, y1, y2, z, z1, z2$ ) — объемный инерционный тензор для ограниченной области;
- VOLUME\_INERTIA** ( $x, x1, x2, y, y1, y2, z, z1, z2, mu$ ) — объемный инерционный тензор инерции для ограниченной области с распределением плотности  $mu(x, y, z)$ ;
- SPHERICAL\_VOLUME** ( $r, r1, r2, th, th1, th2, phi, phi1, phi2$ ) — объем ограниченной области пространства в сферической системе координат с параметрами изменения  $r$  от  $r1$  до  $r2$ ,  $th$  от  $th1(r)$  до  $th2(r)$ ,  $phi$  от  $phi1(r, th)$  до  $phi2(r, th)$ ;
- SPHERICAL\_VOLUME** ( $r, r1, r2, th, th1, th2, phi, phi1, phi2, mu$ ) — интеграл от  $mu(r, th, phi)$  по ограниченной области в сферической системе координат, где  $th$  и  $phi$  — географическая долгота и широта (в радианах);
- CYLINDRICAL\_VOLUME** ( $r, r1, r2, th, th1, th2, z, z1, z2$ ) — объем ограниченной области в цилиндрической системе координат при изменении  $z$  от  $z1$  до  $z2$ ,  $th$  от  $th1(z)$  до  $th2(z)$ ,  $r$  от  $r1(th, z)$  до  $r2(th, z)$ ;

**CYLINDRICAL\_VOLUME****(r, r1, r2, th, th1, th2, z, z1, z2, mu)**— интеграл по ограниченной области в цилиндрической системе координат при заполнении области веществом с плотностью  $\mu(r, th, ph)$ ;**AREA\_OF\_REVOLUTION (y, x, x1, x2)**— площадь ограниченной поверхности, получаемой вращением линии  $y(x)$  вокруг оси  $x$  при изменении  $x$  от  $x_1$  до  $x_2$ ;**VOLUMEY\_OF\_REVOLUTION****(y, x, x1, x2)**— объем фигуры, полученной вращением линии  $y(x)$  вокруг оси  $y$  при изменении  $x$  от  $x_1$  до  $x_2$ .

Эти функции рассчитаны на применение команды SIMPLIFY. Если решение в аналитическом виде найти невозможно, надо использовать команду arproX для получения численного решения.

Приведенные ниже примеры поясняют применение некоторых из этих функций.

1: "Дополнительные интегральные функции"

2: FOURIER ( $x^2$ , x, 0, 1, 2)

$$3: \frac{\cos(4\pi x)}{4\pi} - \frac{\sin(4\pi x)}{2\pi} + \frac{\cos(2\pi x)}{\pi} - \frac{\sin(2\pi x)}{\pi} + \frac{1}{3}$$

4: LAPLACE ( $t^2$ , t, s)

$$5: \int_0^{\infty} t^2 e^{-st} dt$$

6: LAPLACE ( $t^2$ , t, 3)

7: 0.0740740

8: ARC\_LENGTH ( $x^2$ , x, 1, 2)

9: 3.16784

10: SURFACE\_AREA ( $x^2 + y^2$ , x, -1, 1, y, -1, 1)

11: 7.44625

12: AREA\_OF\_REVOLUTION ( $x^2$ , x, 1, 2)

13: 49.4162

В этих примерах возможности функций интегрирования даны выборочно. Тестирование остальных функций можно провести аналогично.

## 3.7. Вычисление специальных математических функций

Средства Derive позволяют вычислять практически все специальные математические функции по алгоритмам, описанным в литературе [30]. Для этого необходима загрузка соответствующих файлов. Их имена и вводимые с их помощью функции указаны ниже. Для ряда функций вычисление реализовано двумя методами — с применением численного интегрирования и разложения их в ряд с числом суммируемых членов  $m$  (или  $m + 1$ ). В последнем случае в названии ряда функций стоит слово *series* (ряд). Для каждого класса функций приведены контрольные примеры.

### 3.7.1. Интегральные показательные функции

Интегральные показательные функции в файле `exp_int.mth` содержат следующие определения:

```
euler_gamma:=0.577215664901532860606512
```

```
EI(x, m):=euler_gamma + LN(x) + SUM(x^n_/(n_*n_!), n_, 1, m)
```

```
LI(x, m):=EI(LN(x), m)
```

```
EN(n, z):=INT(#e^(-z*t_)/t_^n, t_, 1, inf)
```

```
EN_ASYMP(n, z, m):=#e^(-z)/z*SUM(PERM(n+l_-1,l_-1)/(-z)^l_, l_, 0, m)
```

```
E1(z, m):=-euler_gamma-LN(z)-UM((-z)^n_/(n_*n_!), n_, 1, m)
```

```
SI(z):=INT(SIN(t_)/t_, t_, 0, z)
```

```
CI(z):=euler_gamma + LN(z) + INT((COS(t_)-1)/t_, t_, 0, z)
```

Ниже представлено их назначение:

|                          |                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>euler_gamma</b>       | — константа Эйлера (25 точных цифр);                                                                                                  |
| <b>EI(x, m)</b>          | — интегральная показательная функция $\text{INT}(\#e^{-t}/t, t, -x, \text{inf})$ для $x > 0$ , вычисленная по ее разложению в ряд;    |
| <b>LI(x, m)</b>          | — логарифмическая интегральная функция $\text{INT}(1/\text{LN } t, t, 0, x)$ для $x > 1$ , вычисленная по ее разложению в ряд;        |
| <b>EN(n, z)</b>          | — интегральная показательная функция $n$ -го порядка $\text{INT}(\exp(-zt)/t^n, t, 1, \text{inf})$ при $\text{RE}(z) > 0, n \geq 0$ ; |
| <b>EN_ASYMP(n, z, m)</b> | — асимптотическое приближение $\text{EN}(n, z)$ для больших $ z $ , вычисляемое по разложению в ряд, ограниченный $m + 1$ членами;    |
| <b>E1(z, m)</b>          | — показательная функция $E1(z) = \text{EN}(1, z)$ , вычисляемая по ее разложению в ряд;                                               |
| <b>SI(z)</b>             | — интегральный синус $\text{INT}(\text{SIN}(t)/t, t, 0, z)$ ;                                                                         |
| <b>CI(z)</b>             | — интегральный косинус $\text{INT}(\text{COS}(t)/t, t, 0, z)$ для модуля аргумента $z$ , меньшего $\pi$ .                             |

Примеры вычисления этих функций представлены ниже.

```

1: "Интегральные показательные функции"
2: euler_gamma
3: 0.577215
4: EI (0.5, 2)
5: 0.446568
6: LI (0.5, 2)
7: -0.362331 + 3.14159 #i
8: EN (2, 0.5)
9: 0.326643
10: EN_ASYMP (2, 10, 5)
11: 3.81723 10-6
12: E1 (2, 2)
13: -0.270362
14: SI (1)
15: 0.946083
16: CI (1)
17: 0.337404

```

### 3.7.2. Дополнительные вероятностные функции

Дополнительные вероятностные функции, широко применяемые в специальных статистических расчетах, содержатся в файле `probabil.mth`:

**POSHHAMMER (a, x)**

— функция  $(a)x = \text{GAMMA}(a + x)/\text{GAMMA}(a)$ ;

**PSI (z)**

— пси-функция  $\text{DIF}(\text{GAMMA}(z), z)/\text{GAMMA}(z)$  для модуля аргумента  $z$ , не превосходящего  $\pi/2$ ;

**POLYGAMMA (n, z, m)**

— полигамма-функция  $\text{DIF}(\text{PSI}(z), z, n)$ , где  $z/ = 0, -1, -2, \dots$ , вычисляемая по разложению в ряд;

**INCOMPLETE\_GAMMA (z, w)**

— функция  $P(z, w) = \text{INT}(\#e^{-t} * t^{z-1}, t, 0, w)/\text{GAMMA}(z)$ ,  $\text{RE}(z) > 0$ ;

**INCOMPLETE\_GAMMA\_SERIES**

— функция  $P(z, w)$ , вычисляемая по  $(z, w, m)$  разложению в ряд;



|                                              |                                                                                                                           |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <b>BETA</b> (z, w)                           | — бета-функция $B(z, w) = \text{GAMMA}(z) * \text{GAMMA}(w) / \text{GAMMA}(z + w)$ ;                                      |
| <b>INCOMPLETE_BETA</b> (x, z, w)             | — функция $B_x(z, w) = \text{INT}(t^{z-1} * (1-t)^{w-1}, t, 0, x)$ ;                                                      |
| <b>POISSON_DENSITY</b> (k, t)                | — плотность вероятности распределения Пуассона $\#e^{-t} * t^k / k!$ ;                                                    |
| <b>POISSON_DISTRIBUTION</b> (k, t)           | — распределение Пуассона, т. е. функция $\text{SUM}(\#e^{-t} * t^j / j!, j, 0, k)$ ;                                      |
| <b>BINOMIAL_DENSITY</b> (k, n, p)            | — функция $\text{COMB}(n, k) * p^k * (1-p)^{n-k}$ ;                                                                       |
| <b>BINOMIAL_DISTRIBUTION</b> (k, n, p)       | — биномиальное распределение, т. е. функция $\text{SUM}(\text{COMB}(n, j) * p^j * (1-p)^{n-j}, j, 0, \text{MIN}(k, n))$ ; |
| <b>HYPERGEOMETRIC_DENSITY</b> (k, n, p)      | — гипергеометрическое распределение $\text{COMB}(m, k) * \text{COMB}(j-m, n-k) / \text{COMB}(j, n)$ ;                     |
| <b>HYPERGEOMETRIC_DISTRIBUTION</b> (k, n, p) | — функция кумулятивного гипергеометрического распределения;                                                               |
| <b>STUDENT</b> (t, v)                        | — кумулятивная плотность распределения Стьюдента $A(t v)$ ;                                                               |
| <b>F_DISTRIBUTION</b> (f, v1, v2)            | — F-функция кумулятивного распределения $P(f v1, v2)$ ;                                                                   |
| <b>CHI_SQ</b> (u, v)                         | — функция Chi-квадрат распределения $P(u v)$ , $u = \text{Chi}^2$ .                                                       |

Далее поясняется применение этих функций.

1: "Вычисление добавочных функций распределения"

2: POCHHAMMER (2, 0.5)

3: 1.32934

4: PSI (1)

5: -0.577215

6: POLYGAMMA (2, 0.5, 5)

7: -16.8289

8: POISSON\_DENSITY (k, t)

          k LN (t) - t  
          #e  
9: -----  
          k!

10: POISSON\_DENSITY (2, 0.5)

11: 0.0758163

12: HYPERGEOMETRIC\_DENSITY (k, n, m, j)

```

13: -----
 n! m! (j - n)! (j - m)!
 (j + k - m - n)! j! k! (m - k)! (n - k)!
14: HYPERGEOMETRIC_DENSITY (1, 2, 3, 4)
15: 0.5
16: STUDENT (0.5, 1)
17: 0.295169
18: F_DISTRIBUTION (0.5, 1, 2)
19: 0.552786
20: CHI_SQ (1, 1)
21: 0.682690

```

### 3.7.3. Интегралы Френеля

Интегралы Френеля в файле `fresnel.mch` представлены через свои интегральные определения:

```
FRESNEL_SIN(z):=INT(SIN(pi*t_^2/2), t_, 0, z)
```

```
FRESNEL_SIN_SERIES(z, m):=SUM(COS(pi*k_)*(pi/2)^(2*k_ + 1)*z^
(4*k_+3)/((2*k_ + 1)*(4*k_ + 3)), k_, 0, m)
```

```
FRESNEL_COS(z):=INT(COS(pi*t_^2/2), t_, 0, z)
```

```
FRESNEL_COS_SERIES(z, m):=SUM(COS(pi*k_)*(pi/2)^(2*k_)*z^
(4*k_ + 1)/((2*k_)!*(4*k_ + 1)), k_, 0, m)
```

```
FRESNEL_AUX_F(z):=1/(pi*z)-3/(pi^3*z^5)
```

```
FRESNEL_AUX_G(z):=1/(pi^2*z^3)-15/(pi^4*z^7)
```

```
FRESNEL_SIN_ASYMP(z):=1/2-FRESNEL_AUX_F(z)*COS(pi*z^2/2)--
```

```
FRESNEL_AUX_G(z)*SIN(pi*z^2/2)
```

```
FRESNEL_COS_ASYMP(z):=1/2 + FRESNEL_AUX_F(z)*SIN(pi*z^2/2)--
```

```
FRESNEL_AUX_G(z)*COS(pi*z^2/2)
```

Эти функции обозначаются так:

**FRESNEL\_SIN (z)** — синусный интеграл Френеля  $S(z) = \text{INT}(\text{SIN}(\pi * t^2/2), t, 0, z);$

**FRESNEL\_SIN\_SERIES (z, m)** — синусный интеграл Френеля, вычисляемый по разложению в ряд;

**FRESNEL\_COS (z)** — косинусный интеграл Френеля  $C(z) = \text{INT}(\text{COS}(\pi * t^2/2), t, 0, z);$

**FRESNEL\_COS\_SERIES** ( $z, m$ ) — косинусный интеграл Френеля, вычисляемый по разложению в ряд.

Ниже даны примеры вычисления интегралов Френеля.

```

1: "Вычисление интегралов Френеля"
2: FRESNEL_SIN (1)
3: 0.438258
4: FRESNEL_SIN_SERIES (1, 5)
5: 0.438259
6: FRESNEL_COS (1)
7: 0.779894
8: FRESNEL_COS_SERIES (1, 5)
9: 0.779893

```

### 3.7.4. Функции Бесселя и Эйри

Функции Бесселя и Эйри содержатся в файле `bessel.mth`:

**BESSEL\_J** ( $n, z$ ) — функция Бесселя первого рода  $J_n(z)$  при рациональном  $n$ ;

**BESSEL\_J\_ASYMP** ( $n, z$ ) — асимптотическое приближения  $J_n(z)$  для больших  $|z|$ , вычисляемое по разложению в ряд;

**BESSEL\_Y** ( $n, z$ ) — функция Бесселя второго рода  $Y_n(z)$  при дробном  $n$ ;

**BESSEL\_Y\_SERIES** ( $n, z, m$ ) — функция Бесселя второго рода, вычисляемая по разложению в ряд;

**BESSEL\_Y\_ASYMP** ( $n, z$ ) — асимптотическое приближение  $Y_n(z)$  для больших  $|z|$ ;

**SPHERICAL\_BESSEL\_J** ( $n, z$ ) — сферические функции Бесселя первого рода в закрытой форме  $j_n(z)$ ;

**SPHERICAL\_BESSEL\_Y** ( $n, z$ ) — сферические функции Бесселя второго рода в закрытой форме  $y_n(z)$ ;

**AI\_SERIES** ( $z, m$ ) — функция Эйри  $Ai(z)$ , вычисляемая по разложению в ряд;

**BI\_SERIES** ( $z, m$ ) — функция Эйри  $Bi(z)$ , вычисляемая по разложению в ряд.

Ниже показано вычисление функций Бесселя.

```

1: "Вычисление функций Бесселя"
2: BESSEL_J (2, 0.5)
3: 0.0306040

```

- 4: BESSEL\_J (2, 0.5, 2)
- 5: 0.0306040
- 6: BESSEL\_J\_ASYMP (10, 10)
- 7: 0.246760
- 8: BESSEL\_Y\_SERIES (2, 1, 5)
- 9: -1.65068
- 10: BESSEL\_Y (0.2, 0.5)
- 11: -0.700968
- 12: AI\_AUX (1, 5)
- 13: 3.14051
- 14: BI\_AUX (1, 5)
- 15: 1.46967

### 3.7.5. Гипергеометрические функции

Гипергеометрические функции содержатся в файле `hipergeo.mth`:

- KUMMER (a, b, z)** — конфлюэнтная гипергеометрическая функция Куммера  $M(a, b, z)$ ;
- KUMMER\_SERIES (a, b, z, m)** — функция Куммера  $M(a, b, z)$ , вычисляемая по разложению в ряд;
- HYPERGEOMETRIC (a, b, c, z)** — гипергеометрическая функция Гаусса  $F(a, b, c, z)$ ;
- HYPERGEOMETRIC (a, b, c, z, m)** — гипергеометрическая функция Гаусса  $F(a, b, c, z)$ , вычисляемая по разложению в ряд.

В следующих примерах показывается вычисление гипергеометрических функций:

- 1: "Вычисление гипергеометрических функций"
- 2: KUMMER (1, 2, 1)
- 3: 1.71828
- 4: HYPERGEOMETRIC (1, 1, 2, 0.5)
- 5: 1.38629
- 6: HYPERGEOMETRIC\_SERIES (1, 1, 2, 0.5, 5)

```

7: 1.38229
8: HYPERGEOMETRIC_SERIES (1, 1, 2, 0.5, 20)
9: 1.38629

```

### 3.7.6. Эллиптические интегралы

Эллиптические интегралы в файле `elliptic.mth` заданы своими интегральными представлениями:

```

ELLIPTIC_F(phi, m):=INT(1/SQRT(1-m*SIN(t_)^2), t_, 0, phi)
ELLIPTIC_E(phi, m):=INT(SQRT(1-m*SIN(t_)^2), t_, 0, phi)
ELLIPTIC_PI(phi, m, n):=INT(1/((1-n*SIN(t_)^2)*SQRT(1-
-m*SIN(t_)^2)), t_, 0, phi)

```

Они обозначаются так:

**ELLIPTIC\_F (phi, m)** — эллиптический интеграл первого рода  $F(\phi \setminus m)$ ;  
**ELLIPTIC\_E (phi, m)** — эллиптический интеграл второго рода  $E(\phi \setminus m)$ ;  
**ELLIPTIC\_PI (phi, m, n)** — эллиптический интеграл третьего рода  $\Pi(n, \phi \setminus m)$ .

Вычисление эллиптических интегралов демонстрируют следующие примеры:

```

1: "Вычисление эллиптических интегралов"
2: ELLIPTIC_F (0.5, 2)
3: 0.551358
4: ELLIPTIC_E (0.5, 2)
5: 0.456992
6: ELLIPTIC_PI (0.5, 2, 2)
7: 0.687784

```

### 3.7.7. Ортогональные полиномы

Ортогональные полиномы представлены функциями файла `orth_pol.mth`:

**CHEBYCHEV\_T (n, x)** — полином Чебышева первого рода  $T_n(x)$ ;  
**CHEBYCHEV\_U (n, x)** — полином Чебышева второго рода  $U_n(x)$ ;  
**LEGENDRE\_P (n, x)** — полином Лежандра  $P_n(x)$ ;  
**ASSOCIATED\_LEGENDRE\_P (n, m, x)** — полином Лежандра  $P_n^m(x)$ ;  
**HERMITE\_H (n, x)** — полином Эрмита  $H_n(x)$ ;  
**HERMITE\_HE (n, x)** — ассоциированный полином Эрмита  $HE_n(x)$ ;  
**WEBER\_D (n, x)** — цилиндрическая параболическая функция Вебера  $D_n(x)$ ;

- GENERALIZED\_LAGUERRE**(n, a, x) — полином Лагерра в общей форме  $L_n(n, a, x)$  при  $a = 0$  — ординарная форма;
- JACOBI\_P**(n, a, b, x) — полином Якоби  $P_n^a(a, b)(x)$ ;
- GEGENBAUER\_C**(n, a, x) — ультрасферический полином Гегенбауэра  $C_n^a(x)$ .

Задачи на вычисление ортогональных полиномов показаны в следующем примере:

- 1: "Вычисление ортогональных полиномов"
- 2: CHEBYSHEV\_T (3, x)
- 3:  $x (4 x^2 - 3)$
- 4: CHEBYSHEV\_U (3, x)
- 5:  $4 x (2 x^2 - 1)$
- 6: LEGENDRE\_P (3, x)
- 7:  $0.5 x (5 x^2 - 3)$
- 8: HERMITE\_H (3, x)
- 9:  $4 x (2 x^2 - 3)$
- 10: WEBER\_D (3, x)
- 11: 
$$- \#e^{x^2/4} \operatorname{DIF} \left[ \#e^{-\left(\sqrt{2} x/2\right)^2}, \frac{\sqrt{2} x}{2}, 3 \right]$$
- 12: GENERALIZED\_LAGUERRE (3, 1, x)
- 13:  $- 0.166666 (x^3 - 12 x^2 + 36 x - 24)$

### 3.7.8. Дзета-функция, полилогарифм и дилогарифм (файл zeta.mth)

Дзета-функция, полилогарифм и дилогарифм содержатся в файле zeta.mth:

- ZETA**(s, m) — дзета-функция Римана  $SUM(k^{-s}, k, 1, \text{inf})$ , вычисляемая по разложению в ряд;
- HURWITZ\_ZETA**(s, a, m) — Гурвица функция  $SUM((k+a)^{-s}, k, 0, \text{inf})$ , вычисляемая по разложению в ряд;
- POLYLOG**(n, z, m) — полилогарифм  $\operatorname{Lin}(z)$ , вычисляемый по разложению в ряд;
- LERCH\_PHI**(z, s, a, m) — пси-функция  $\operatorname{Phi}(z, s, a)$ , вычисляемая по разложению в ряд;
- DILOG**(x) — дилогарифм  $\operatorname{INT}(\operatorname{LN}(t)/(t-1), t, 1, x)$ .

Следующие примеры иллюстрируют вычисление этих функций:

```

1: "Вычисление функций файла zeta.mth"
2: ZETA (0.25, 5)
3: -0.812045
4: HURWITZ_ZETA (0.25, 1, 5)
5: -0.812276
6: POLYLOG (2, 0.25, 5)
7: 0.267644
8: LERCH_PHI (0.5, 1, 1, 5)
9: 1.38229
10: DILOG (x)
11: - ∫1x $\frac{\text{LN}(t)}{t-1}$ dt
12: DILOG (0.5)
13: 0.582240

```

Из приведенного описания видно, что библиотеки Derive содержат широкий набор специальных математических функций, который вполне удовлетворяет потребности в вычислениях таких функций, возникающие у большинства научных работников и инженеров. Практически заданы определения большинства из основных функций, входящих в энциклопедический справочник по специальным функциям [30]. Некоторые из представлений специальных функций на языке программирования системы Derive были приведены при их описании. С другими можно ознакомиться, просмотрев тексты соответствующих файлов \*.mth с помощью текстовых редакторов или выверов.

К недостаткам Derive надо отнести невозможность ряда аналитических операций со специальными математическими функциями, поскольку в ходе этих операций система возвращает не имена соответствующих функций, а их громоздкие интегральные или иные представления. Например, производные от функций Бесселя — те же функции Бесселя, но иного порядка. Derive вместо этого выводит их громоздкие представления. Иногда можно вручную записать нужное выражение, но это не слишком удобно. Тут более мощные системы Maple V или Mathematica 2/3 куда более предпочтительны.

## 3.8. Графические функции

Файл graphics.mth содержит большое количество функций для специальных графических вычислений. Это функции, создающие матрицы поверхностей, обеспечивающие поворот фигур вокруг осей на заданный угол, функции для

изометрической проекции и многие другие. Для построения фигур с помощью таких функций используется окно двумерной графики.

Функции файла `graphics.mth` следующие:

- ISOMETRIC (v)** — матрица двумерной изометрической проекции трехмерного координатного вектора  $v = [x(t), y(t), z(t)]$ , представляющего трехмерную фигуру (используется с командой `Plot`);
- ROTATE\_X (phi)** — матрица  $A$ , такая, что  $A.[x, y, z]$  поворачивает фигуру на угол  $\phi$  относительно оси  $x$ ;
- ROTATE\_Y (phi)** — матрица  $A$  такая, что  $A.[x, y, z]$  поворачивает фигуру на угол  $\phi$  относительно оси  $y$ ;
- ROTATE\_Z (phi)** — матрица  $A$  такая, что  $A.[x, y, z]$  поворачивает фигуру на угол  $\phi$  относительно оси  $z$ ;
- ISOMETRICS** — матрица поверхности  $[x(s, t), y(s, t), z(s, t)]$ ,  $s = s_0 \dots s_m$ ,  $t = t_0 \dots t_n$  (используется команда `arproX` при работе команды `Plot`);
- CYLINDER (r, theta, z)** — трехкоординатный вектор, определяющий декартовы координаты цилиндра с радиусом  $r$  от оси  $z$ ;
- SPHERE (r, theta, phi)** — трехкоординатный вектор, определяющий декартовы координаты точки, чьи сферические координаты заданы радиусом  $r$ , долготой  $\theta$  и широтой  $\phi$ ;
- TORUS (r, theta, phi)** — трехкоординатный вектор, определяющий декартовы координаты точки с тороидальными координатами  $(r, \theta, \phi)$ , где  $r$  — радиус от начала координат в центр трубы;
- CONE (phi, theta, z)** — трехкоординатный вектор, определяющий декартовы координаты круглого конуса с вершиной в начале координат, поверхность которого составляет с осью  $z$  угол  $\phi$ ;
- RAYS (w, z, z00, zmn, m, n)** — матрица  $w$  векторов для  $z = x_0 + \#i * y_0 \dots x_m + \#i$ , отображаемая в графическом окне в виде набора лучей на прямоугольной сетке, с длиной и направлением, соответствующим значению  $w$ ;
- HORIZONTALS** — матрица, представляющая в двумерном окне набор точек, где каждый набор значений  $w$  соединен отрезками прямых;
- ARCS (w, z, r0, rm, m, th0, thm, n)** — матрица, представляющая в двумерном окне набор точек, где каждый ряд значений соединен сегментами.

Кроме того, в этом файле определена переменная `axes`, значением которой является матрица  $3 \times 2$ , параметрически задающая изометрическую проекцию осей  $X$ ,  $Y$  и  $Z$  в трехмерной Декартовой системе координат. Эта переменная используется для графического построения осей.



Примеры вычисления некоторых графических функций представлены ниже.

1: "Некоторые графические функции"

2: axes

3: 
$$\begin{pmatrix} -t_ & -0.5 t_ \\ t_ & -0.5 t_ \\ 0 & t_ \end{pmatrix}$$

4: ROTATE\_Z (a)

5: 
$$\begin{pmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

6: CYLINDER (1, t, z)

7: [COS (t), SIN (t), z]

8: CONE (a, t, z)

9: [z COS (t) SIN (a), z SIN (t) SIN (a), z]

10: RE\_IM (a + #i b)

11: [a, b]

Большинство функций файла graphics.mth служит в различных графических построениях для вычисления координат точек, необходимых для их преобразования, например поворота, или вычисления их геометрических параметров. Эти функции не означают прямого построения по ним каких-либо графиков, они просто реализуют математические расчеты, встречающиеся в аналитической геометрии и в математических методах обработки изображений.

В качестве примера применения графических функций для построения графиков рассмотрим построение в трехмерном пространстве спирали на фоне координатных осей. Прежде всего следует загрузить файл graphics.mth, используя команды TRANSFER, LOAD и UTILITY. Затем с помощью команды AUTHOR введите выражения

**axes**

и

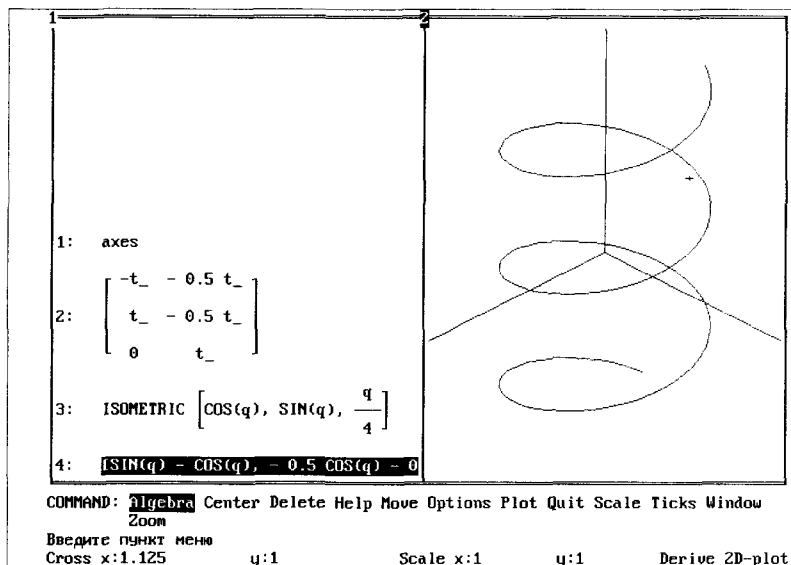
**ISOMETRIC ([COS(q), SIN(q), q/4]).**

Используя команду arроX, получите их значения. На рис.3.1 они представлены в левом окне; правое окно создано для вывода графиков. Позаботьтесь о том, чтобы обычные оси оказались невидимыми. Для этого задайте их цвет, идентичный цвету фона. Выделив матрицу, полученную от исполнения

выражения axes в диапазоне значений переменных координатных осей от 0 до 9, получим три оси на рис. 3.1. Затем, выделив выражение в строке 4, построим спираль.

Поупражняйтесь сами, пока не получите полного соответствия с изображением, показанным на рис. 3.1.

**Рис. 3.1**  
Построение  
объемной  
спирали



## 3.9. Паде-аппроксимация

Одной из наиболее точных аппроксимаций нелинейных зависимостей вида  $y(x)$  является аппроксимация дробно-рациональными функциями — Паде-аппроксимация. При ней зависимость  $y(x)$  заменяется функцией, представляющей собой отношение полиномов. Паде-аппроксимация может использоваться для приближения даже функций с разрывами (например, функции тангенса), когда обычная аппроксимация полиномом (степенным многочленом) дает явно неудовлетворительные результаты.

Реализация такой аппроксимации в среде Derive возможна после загрузки файла внешнего расширения `arproh.mth`. Он содержит (наряду со вспомогательными функциями) функцию PADE:

$$CF(a, m, n) := \text{VECTOR}(\text{VECTOR}(\text{ELEMENT}(a, m - n + i + j), j, 1, n + 1), i, 1, n)$$

$$DV(a, m, n) := \text{ELEMENT}(\text{ROW\_REDUCE}(CF(a, m, n)), n + 1)$$

$$NV(a, u, m, n) := \text{VECTOR}(\text{ELEMENT}(a, k + 1) - \text{SUM}(\text{ELEMENT}(u, n + 1 - i) * \text{ELEMENT}(a, k + 1 - i), i, 1, k), k, 1, \text{MIN}(m, n))$$

$$P3(a, v, u, x, c, m, n) := (\text{ELEMENT}(a, 1) + \text{SUM}(\text{ELEMENT}(v, k) * (x - c)^k, k, 1, m)) / (1 - \text{SUM}(\text{ELEMENT}(u, n + 1 - k) * (x - c)^k, k, 1, n))$$

$$P2(a, u, x, c, m, n) := P3(a, NV(a, u, m, n), u, x, c, m, n)$$

$$P1(a, x, c, m, n) := P2(a, DV(a, m, n), x, c, m, n)$$

$$PADE(y, x, x0, n, d) := P1(VECTOR(LIM(DIF(y, x, m\_), x, x0)/m\_!, m\_ , 0, n+d+1), x, x0, n, d)$$

Рациональная Паде-аппроксимация с помощью этой функции реализуется обращением:

$$PADE(y, x, x0, n, d)$$

и дает аппроксимацию вблизи  $x = x0$  отношением двух многочленов, причем  $n$  — степень числителя и  $d$  — степень знаменателя ( $n = d$  или  $d - 1$ ). Приведенный ниже пример иллюстрирует технику проведения рациональной аппроксимации.

1: "Паде-аппроксимация"

$$2: PADE (x^2 + 1, x, 1, 3, 3)$$

$$3: x^2 + 1$$

$$4: PADE (SIN (x), x, 1, 2, 2)$$

$$5: - \frac{0.00306523 \cdot 10^0 (3.94074 \cdot 10^3 x^2 - 1.27412 \cdot 10^{11} x + 9.35916 \cdot 10^8)}{3.21290 \cdot 10^7 x^2 - 9.08822 \cdot 10^7 x + 3.75921 \cdot 10^8}$$

В строке 2 этого примера задана Паде-аппроксимация для функции  $(x^2 + 1)$ . Поскольку эта функция сама по себе полином, то функция PADE выдала исходную функцию. Зато аппроксимация функции  $\sin(x)$  ведет к выдаче отношения полиномов в полном виде.

При проведении аппроксимации большой интерес представляет сравнение результатов различных видов аппроксимации с представлением результатов в графическом виде. На рис. 3.2 показана аппроксимация разрывной функции  $TAN(x)$  двумя способами: с помощью степенного полинома (полученного разложением функции относительно точки  $x0 = 1$  в ряд Тейлора) и с помощью аппроксимации дробно-рациональной функцией с помощью Паде-аппроксимации.

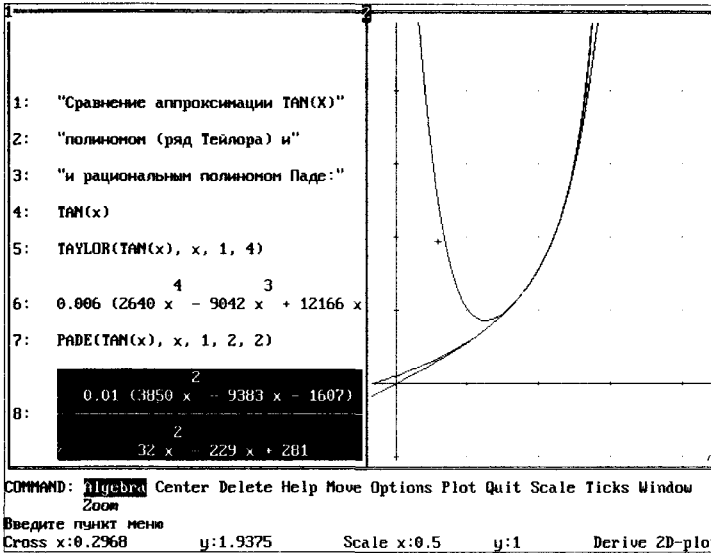
Необходимые для аппроксимации действия показаны в левом окне. Там же находятся результаты аппроксимации. Выделив функцию  $TAN(x)$  и результаты ее аппроксимации, можно построить графики функций. Они представлены на рис. 3.2 в правом окне.

При интерпретации нескольких графиков возникают определенные проблемы по их выделению. Derive строит графики с разными цветами линий, поэтому их можно идентифицировать по цвету. Однако при печати графиков на обычных принтерах (не цветных) этот способ не годится.

Один из способов идентификации заключается в словесном описании. Например, на рис. 3.2 график полиномиальной аппроксимации рядом Тейлора помечен крестиком-меткой. График ниже — Паде-аппроксимация, еще ниже —

Рис.3.2

Пример  
аппроксимации  
функции  $\text{TAN}(x)$   
двумя методами



просто график функции  $\text{TAN}(x)$ . Нетрудно заметить, что аппроксимация рядом Тейлора дает большие ошибки при приближении графика  $\text{TAN}(x)$  не только к точкам разрыва, но и к нулю. Паде-аппроксимация в этом отношении намного лучше. С помощью графического редактора PaintBrush можно нанести на графики наглядные указания и комментарии. В версии Derive под Windows предусмотрено нанесение на график в любом месте текстовых комментариев.

## 3.10. Прочие функции

В файле misc.mth система Derive содержит определения ряда различных функций, которые трудно отнести к описанным ранее группам. Эти дополнительные функции перечислены ниже.

**MOD (a, b)**

— деление  $a$  на  $b$  по модулю (дает неотрицательный остаток от  $a/b$ );

**FLOOR (a, b)**

— наибольшее целое меньшее или равное  $a/b$ ;

**RATIO\_TEST (t, n)**

— если значение данной функции больше 1, то  $\text{SUM}(t, n, a, \text{inf})$  дает сумму конвергенций, а при значении функции 1 — сумму дивергенций;

**LIM2 (u, x, y, x0, y0)**

— предел  $u(x, y)$  при  $[x, y]$ , стремящемся к  $[x_0, y_0]$  по всей линии наклона;

**LEFT\_RIEMANN (u, x, a, b, n)**

— левая сумма Римана для  $\text{INT}(u, x, a, b)$  и для  $n$  прямоугольников;

**INVERSE (y, x)**

— инвертирование  $y(x)$  по отношению к  $x$ ;

**INT\_PARTS (u, dv, x)**

— интегрирование по частям  $\text{INT}(u \cdot dv, x)$ ;

**INT\_SUBST (y, x, g)**

— действие  $\text{SIMPLIFY INT}(y, x)$  с использованием подстановки  $x \rightarrow g(x)$ ;

**PROVE\_SUM (t, k, a, n, s)**

— дает  $[0, 0]$ , если  $\text{SUM}(t, k, a, n) = s$ .

Ниже представлены примеры вычислений с этими функциями.

1: "Вычисление функций файла misc.mth"

2: MOD (10, 3)

3: 1

4: FLOOR (a, b)

5: 
$$\frac{\text{ATAN} \left( \frac{\text{COT} \left( \frac{\pi a}{b} \right)}{\right)}{\pi} + \frac{2a - b}{2b}$$

6: FLOOR (10, 3)

7: 3

8: INVERSE ( $x^2$ , x)

9: - SQRT (x)

10: RATIO\_TEST (t, n)

11: 1

12: LIM2 ( $x^2 + y^2$ , x, y, 1, 1)

13: 2

14: LEFT\_RIEMANN ( $x^2$ , x, 1, 2, 5)

15: 2.84

16: NTH\_PRIME (5)

17: 11

18: NTH\_PRIME (6)

19: 13

Итак, выше были описаны все функции библиотек внешних расширений системы Derive 2.02. В зависимости от версий системы и условий поставки набор функций может отличаться от описанного. В целом можно сделать вывод, что Derive обладает весьма широкой библиотекой внешних расширений. Пользователь, разумеется, может составить и свои библиотеки внешних расширений и тем самым обеспечить дальнейшую адаптацию системы к решению своих специфических задач.

## 3.11. Описание вспомогательных файлов

Помимо библиотечных файлов в систему Derive входит ряд вспомогательных, в основном демонстрационных файлов. Файл `english.mth` является библиотечным файлом, содержащим определения английских физических величин: фута, мили, ярда и др. Еще один файл — `metric.mth` — содержит метрические единицы, выражаемые в частях метра, грамма и секунды.

Начиная с версии Derive 2.60, система содержит файл `taylor.mth` с функциями для решения в виде ряда Тейлора дифференциальных уравнений второго порядка, решения трансцендентных уравнений в неявном виде, вычисления обратных функций и проведения обычной полиномиальной аппроксимации (степенным многочленом). Эти новые функции представлены ниже (информация из файла `taylor.doc`).

`TAYLOR_ODE2` ( $r, x, y, v, x_0, y_0, v_0, n$ ) упрощается к ряду Тейлора  $n$ -й степени, являющемуся решением уравнения  $y'' = r(x, y, v)$ . Здесь  $v$  обозначает  $y'$ , а  $v_0$  — начальное значение  $v$  при  $x = x_0$ . Например,

`TAYLOR_ODE2 (2*x*v + x^2*y + 3*x, x, y, v, 0, 0, 1, 5)`

упрощается к

$$\frac{3x^5}{10} + \frac{5x}{6} + x^3$$

Вы можете проверить это приближенное решение, подставив его в неупрощенную разность двух частей дифференциального уравнения вместо  $y$  и производную от этого выражения вместо  $v$ , а затем определить, равняется ли ряд Тейлора третьего порядка от этой разности нулю. Здесь используется ряд третьей, а не пятой степени, потому что вторая производная понижает степень ряда на 2.

`TAYLOR_ODE2` упрощается к знаку?, если  $r$  недостаточно дифференцируемо для конечного ряда  $n$ -й степени. Если это произошло, попробуйте положительное целое  $n$  меньшей величины.

`TAYLOR_SOLVE` ( $u, x, y, x_0, y_0, n$ ) упрощается к ряду Тейлора  $n$ -й степени, являющемуся решением  $y(x)$  уравнения  $u(x, y) = 0$ , если известно, что  $u(x_0, y_0) = 0$ . Например, невозможно в общем случае точно решить относительно  $y$  трансцендентное уравнение  $\sin y + y + x \#e^x = 0$ , а дополнительный нечисловой параметр  $x$  делает невозможным общее численное решение. Однако `TAYLOR_SOLVE` раскрывает любое точное решение для конкретного значения  $x$  в конечный ряд, полезный в окрестности этой точки  $x$ , при условии, что такой конечный ряд существует. Например, если  $y_0 = 0$  при  $x_0 = 0$  — очевидное решение для одного конкретного значения  $x$ , то для получения решения в виде конечного ряда в окрестности этой точки необходимо упростить

`TAYLOR_SOLVE (sin y + y + x*#e^x, x, y, 0, 0, 3)`

Тогда вы получите

$$\frac{25x^3}{96} - \frac{x^2}{2} - \frac{x}{2}$$

`TAYLOR_INVERSE` ( $u, x, y, y_0, n$ ) упрощается к ряду Тейлора, аппроксимирующему функцию, обратную  $y = u(x)$ , в окрестности  $y = u(x_0)$ . Она полез-

на тогда, когда  $u(x)$  не может быть точно инвертирована в замкнутой форме. Например,

TAYLOR\_INVERSE ( $x * e^x$ ,  $x$ ,  $y$ ,  $0$ ,  $3$ )

упрощается к

$$\frac{3y^3}{2} - y + y^2$$

POLY\_INTERPOLATE ( $A$ ,  $x$ ) упрощается к полиному от переменной  $x$ , который интерполирует координатные пары  $[x, y]$  из двухстолбцовой матрицы  $A$ . Если POLY\_INTERPOLATE применима, то обычно она предпочтительнее встроенной функции FIT. Например,

POLY\_INTERPOLATE ( $[0, 0]$ ,  $[1, 1]$ ,  $[2, 2]$ ,  $[3, 0]$ ,  $x$ )

упрощается к

$$\frac{x^2(3-x)}{2}$$

В файл number.mth добавлена функция MERSENNE( $n$ ), возвращающая  $n$ -е число Мерсенна.

Ряд демонстрационных файлов имеет расширение .dmo. С этими файлами полезно ознакомиться при самостоятельном освоении системы:

- arith.dmo** — типовые арифметические операции, работа с рациональными и комплексными числами, точные вычисления;
- algebra.dmo** — алгебраические преобразования, действие команд Expand, Factor и Simplify;
- calculus.dmo** — вычисление пределов и производных функций, вычисление неопределенных, определенных и несобственных интегралов, разложение функций в ряд Тейлора, нахождение сумм и произведений членов рядов;
- function.dmo** — вычисление алгебраических функций и функций комбинаторики, статистических функций;
- trig.dmo** — операции с тригонометрическими и обратными тригонометрическими функциями;
- matrix.dmo** — операции с матрицами, демонстрация действий с матричными функциями.

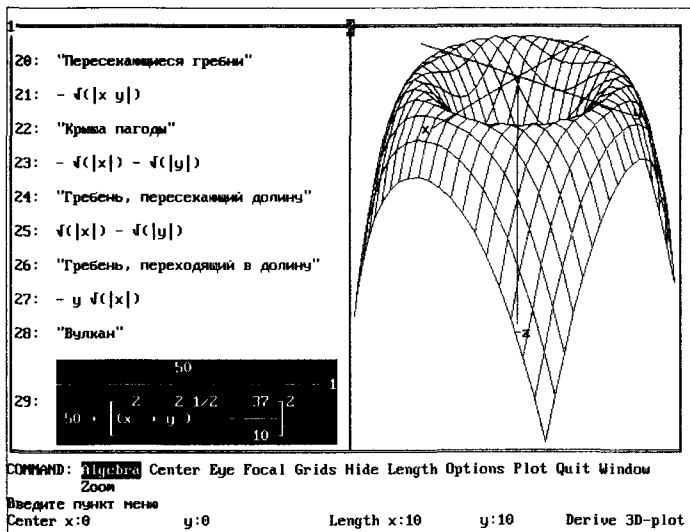
Как уже отмечалось, эти файлы загружаются командами TRANSFER LOAD DEMO и автоматически запускаются на исполнение. Кроме того, в системе эти файлы дублируются файлами с расширением .mth, хотя они и не являются библиотечными. Указанные демонстрационные файлы позволяют пользователю самостоятельно опробовать любой их фрагмент. В систему входят также три демонстрационных файла графики:

- plot2d.mth** — демонстрационная двумерная графика в декартовой системе координат;
- plotpara.mth** — демонстрационная графика для параметрически заданных функций;
- plot3d.mth** — демонстрационная трехмерная графика.

На рис. 3.3 показана работа с загруженным файлом plot3d. В левом окне виден текст документа — комментарии и математические выражения для по-

Рис. 3.3

Пример работы  
с демонстрацион-  
ным файлом  
plot3d



строения трехмерных поверхностей. Одно из выражений выделено, и в правом окне построен график соответствующей поверхности. Для его построения достаточно дважды нажать клавишу P (две команды PLOT).

С демонстрационными файлами и их работой весьма полезно ознакомиться тем пользователям системы, которым она необходима для графических работ. Каждый файл содержит множество примеров графических построений, некоторые из них оригинальны и позволяют по-новому взглянуть на математический синтез кривых.

Кроме указанных файлов, в состав конкретной поставки Derive могут входить и другие файлы. С ними несложно разобраться самостоятельно, тем более, что к таким файлам прилагаются файлы с документацией, подробно описывающей содержащиеся в них функции.

В целом файлы расширения Derive содержат многие сотни новых функций, реализующих довольно сложные и практически полезные алгоритмы. Пользователь может выбрать любые из них для составления своей собственной библиотеки расширений Derive. Это целесообразно, поскольку весьма редко какой-либо библиотечный файл используется полностью, а загрузка неиспользуемых функций оказывается просто бесполезной...



# Глава 4. Математическая система Derive 4.02 под WINDOWS

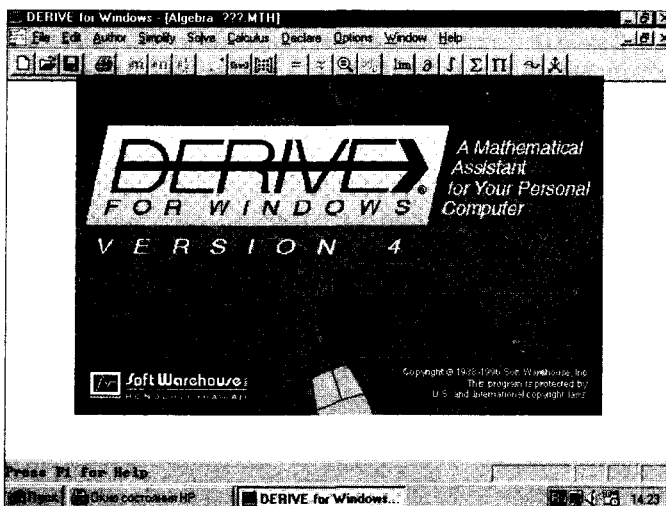
## 4.1. Пятиминутное знакомство с Derive под Windows

### 4.1.1. Запуск Derive под Windows

Математическая система Derive 4.0 под Windows — новый шаг в разработке систем класса Derive. Сохранив все возможности прежних версий этих систем под MS-DOS, новая версия приобрела наиболее важные достоинства приложений под операционные системы класса Windows: намного более удобный и наглядный интерфейс, мощную справочную базу данных, удобные средства ввода и вывода, возможность работы с многими окнами с меняющимися размерами и приложениями. Если работа с версиями под MS-DOS была зачастую примитивной, то с версией 4.02 она стала удобной, простой и более интересной.

При запуске системы появляется основное окно, на фоне которого видна красочная заставка системы (см. рис. 4.1). Хотя система была создана незадолго до появления операционной системы Windows 95, возможность работы в ней предусмотрена, и вид графического интерфейса системы соответствует типовому для приложений Windows 95. В дальнейшем описание относится к системе, работающей в среде Windows 95.

**Рис. 4.1**  
Основное окно системы Derive 4.02 под Windows



В основном окне видны строка заголовка, строка главного меню, сверху панель инструментов и внизу строка состояния системы.

### 4.1.2. Правила работы с мышью

Версии Derive под MS-DOS не предусматривали работу с мышью, и это было их серьезным недостатком. В версии под Windows он, естественно, отсутствует — было бы очень странно, если бы какое-либо приложение под Windows не

управлялось мышью. Сразу же отметим, что здесь предусмотрена краткая справка по элементам интерфейса (в основном по кнопкам панелей инструментов) в виде небольшой бирки с желтым фоном и пояснением, всплывающим, когда курсор мыши приостанавливается на элементе интерфейса. Кроме того, с помощью мыши можно управлять позициями главного меню, размерами и положениями окон, выделением элементов математических выражений и пр.

Надо особо оговорить правила работы с графическим манипулятором — мышью. Она позволяет активизировать позиции меню, нажимать кнопки диалоговых и установочных окон, перемещать эти окна по экрану и изменять их размеры. Правила работы с мышью традиционны для всех Windows-приложений:

- нажать кнопку — установить на кнопку курсор мыши и нажать и отпустить левую ее клавишу;
- активизировать позицию меню или подменю — установить на ней курсор мыши и нажать один раз ее левую клавишу;
- исполнить команду в меню — установить на ее позиции курсор мыши и дважды быстро щелкнуть ее левой клавишей;
- выделить выражение или его часть — установить на них курсор мыши и щелкнуть (или пощелкать) ее левой клавишей;
- выделить часть изображения — установить после соответствующей команды курсор мыши на нужное место и, нажав и удерживая ее левую клавишу, начать ее перемещать по коврику или поверхности стола — появится пунктирный прямоугольник, задающий область копирования.

При нажатии правой клавиши мыши на экран выводится контекстно-зависимое подменю с допустимыми для текущей ситуации командами.

Курсор мыши обычно имеет вид наклонной большой стрелки. Но он может менять свой вид. Например, при загрузке файлов он имеет вид песочных часов, при зацеплении за края окон — вид стрелок, указывающих направления расширения или сужения окон, и т. д. Вряд ли стоит описывать все варианты модификации курсора мыши, они разработаны не в Derive, а в Windows и к тому же интуитивно понятны.

### 4.1.3. Начало работы

Derive под Windows спроектирована так, что начать работу с системой можно буквально в течение пяти минут.

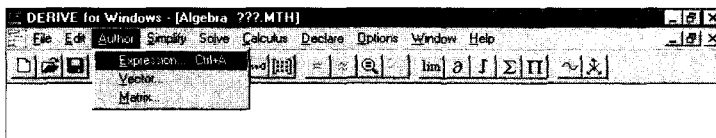
Как и Derive под MS-DOS, новая версия системы реализует общение с пользователем по схеме “вопрос — ответ”. Если вас что-то интересует, задайте системе вопрос на понятном ей математически ориентированном входном языке и укажите, что вы хотите получить. Система даст ответ, для получения которого вам самому, возможно, пришлось бы затратить многие часы, дни и даже месяцы работы. Но вначале узнаем, как Derive выполняет простейшие вычисления и строит графики функций.

Для проведения вычислений надо активизировать позицию главного меню Author. Для этого можно вывести световое выделение на эту позицию, используя клавиши перемещения курсора по горизонтали. Забудьте все, что вы делали в предыдущих версиях Derive под MS-DOS! Здесь все операции с меню и курсором мыши подчиняются типовым правилам, принятым для приложений под Windows. Вы также можете активизировать нужную позицию меню, нажав одновременно клавишу Alt и клавишу, выделенную подчеркнутой буквой — в нашем случае это A.

При активизации позиции Author появится подменю, показанное на рис. 4.2.

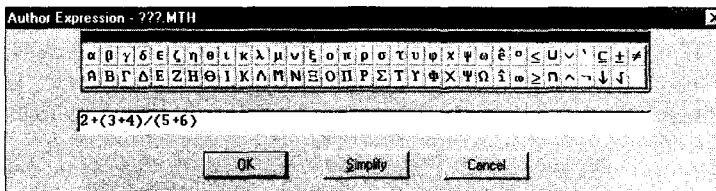
Пока нас в нем интересует только позиция Expression... Ctrl + A. Она служит для вызова диалогового окна ввода математических выражений, которое показано на рис. 4.3. Заметим, что три точки после той или иной команды главного меню указывают на то, что эта команда выполняется не сразу, а выводит соответствующее диалоговое окно. Комбинация “горячих клавиш” Ctrl + A служит для быстрого ввода команды.

**Рис. 4.2**  
Окно Derive с подменю позиции Author



Диалоговое окно ввода выражений позволяет в режиме строчного редактора (по правилам, аналогичным описанным для версии под MS-DOS) вводить математические выражения. Однако для упрощения их ввода в диалоговое окно добавлен новый удобный элемент — панель со специальными математическими символами и греческими буквами. Любой символ можно вставить в выражение, просто указав на него курсором мыши и нажав ее левую клавишу. Это намного повышает оперативность ввода.

**Рис. 4.3**  
Диалоговое окно ввода выражений



Тут, однако, надо сразу заметить, что Derive под Windows работает только с одним специальным набором символов. Он вводится при установке системы и содержит помимо букв латинского алфавита (строчных и прописных) все необходимые специальные символы. Букв русского алфавита в этом наборе нет, а попытка подменить его набором с русскими буквами ведет к уже знакомым из версий под MS-DOS неприятностям: начинают искажаться математические символы. Поскольку это недопустимо, впредь мы не будем пользоваться русскоязычными комментариями в документах Derive. Позже будет описан простой прием, позволяющий избавиться от этого недостатка и готовить документы с любыми шрифтами.

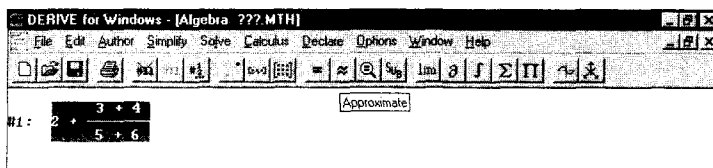
На рис. 4.3 показано введенное в строку выражений арифметическое выражение. Редактирование введенных выражений в новой версии намного проще и естественней, чем в прежних версиях, поскольку для перемещения текстового маркера (вертикальной черточки) используются стандартные клавиши перемещения курсора и весь джентльменский набор средств строчного редактирования, известный всем пользователям Windows. Кстати, поэтому мы не будем детально описывать общепринятые приемы работы с Windows — этого не позволяет объем книги, да и ее тема. В конце концов, книга посвящена только системе Derive, а не описанию операционных систем класса Windows.

Внизу диалогового окна ввода расположены три важные клавиши:

- OK** — фиксация ввода;
- Simplify** — упрощение (пока это касается только исполнения) введенного выражения;
- Cancel** — отказ от ввода выражения и переход в окно выражений.

Итак, если вы ввели выражение и нажали кнопку ОК, это выражение появится в окне выражений (см. рис. 4.4) и будет выделено цветом — синим цветом фона и ярко-белым символом самого выражения. Если нажать кнопку Simplify, выражение будет вычислено и появится в вычисленном виде. Ну а если нажать кнопку Cancel, то набранное выражение просто не будет выведено в окно выражений. Все введенные выражения и результаты их вычислений располагаются на отдельных строчках и последовательно нумеруются знаком вида #N, где N — номер строки.

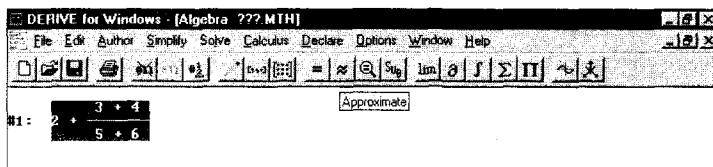
**Рис. 4.4**  
Пример ввода набранного в окне ввода арифметического выражения



Для вычисления выражений Derive обладает рядом возможностей. Одна из них — исполнение команд Basic... (основная) или Approximate... в позиции Simplify главного меню. При этом появляются диалоговые окна, в которых достаточно лишь нажать кнопку ОК.

Но есть и более простой способ выполнения вычислений с применением кнопок быстрого управления системой, расположенных на панели инструментов, прямо под главным меню. Кнопка со знаком = выполняет операцию Simplify, а кнопка со знаком  $\approx$  — операцию Approximate. На рис. 4.5 показан результат

**Рис. 4.5**  
Пример вычисления набранного арифметического выражения



вычисления заданного выражения. Обратите внимание, что под кнопкой  $\approx$  находится начало бирки с полным названием кнопки; как уже говорилось, она появляется, если удерживать курсор мыши на кнопке в течение двух-трех секунд. Позже мы опишем, как управлять форматом вывода результатов вычислений.

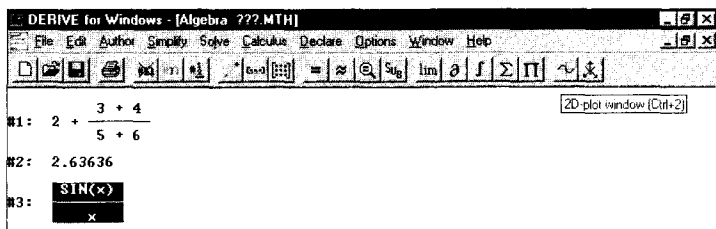
В отношении порядка (иерархии) вычисления отдельных частей математических выражений действуют общепринятые правила приоритета: чем сложнее функция в выражении, тем ранее она вычисляется. Так, умножение и деление выполняются раньше сложения и вычитания, еще раньше вычисляются элементарные функции и т. д. Приоритет легко задавать вводом скобок.

#### 4.1.4. Графическая визуализация вычислений

Позже мы рассмотрим многие куда более сложные примеры вычислений с помощью Derive (в том числе аналитические). А пока обратимся еще к одной первостепенной задаче — построению графиков функций, заданных матема-

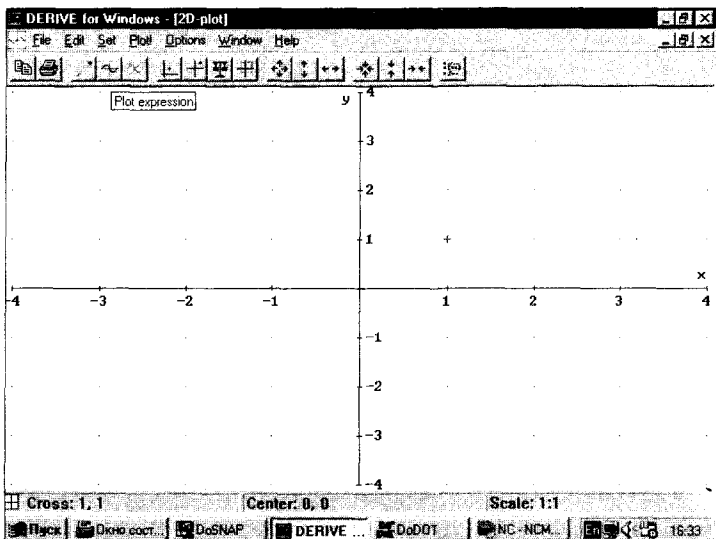
тическими выражениями. Для этого введем еще одно выражение  $\text{SIN}(x)/x$  (см. рис. 4.6). Оно выбрано не случайно — многие программы построения графиков функций искажают построение этой функции в окрестности  $x = 0$ . Они не “догадываются”, что в настоящем случае нет деления на 0, а получается неопределенность  $0/0$ , дающая значение функции 1.

**Рис. 4.6**  
Подготовка  
к построению  
графика функции  
 $\text{SIN}(x)/x$



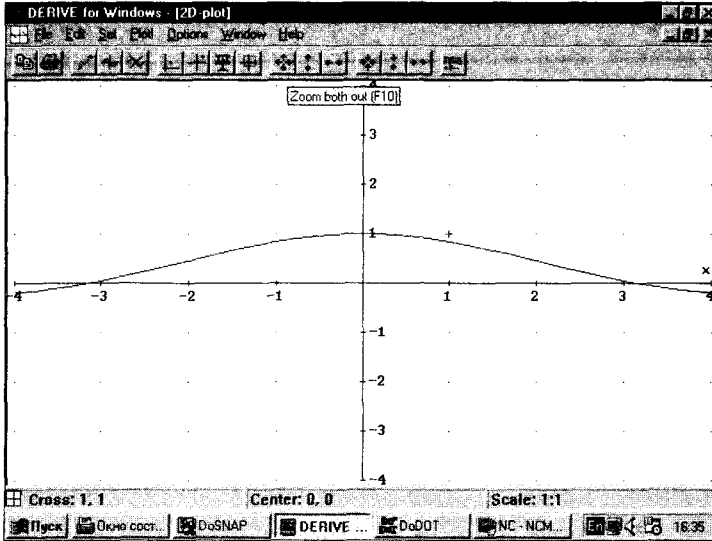
Вероятно, вы уже поняли, что для построения графика функции надо открыть графическое окно, например, с помощью кнопки с изображением графика в декартовой системе координат (вы легко найдете его на рис. 4.6 над началом бирки с надписью “2D-Plot Windows (Ctrl + 2)”). Итак, активизировав эту кнопку установкой на ней курсора мыши и нажатием ее левой клавиши, вы откроете графическое окно, показанное на рис. 4.7.

**Рис. 4.7**  
Графическое  
окно для  
построения  
графика функции



Самого графика в окне пока нет, но уже есть координатные оси и разметка точками площади будущего рисунка. Позже будет описано, как разметить площадь линиями и как изменить формат графика. Однако надо отметить, что Derive автоматически устанавливает достаточно приемлемый формат графика и позволяет быстро (см. клавиши со стрелками на панели инструментов графического окна) изменять его размеры по горизонтали, по вертикали или сразу по обоим осям. Для построения графика надо вновь нажать клавишу с изображением графика (см. рис. 4.8). На рис. 4.7 под этой клавишей видна ее бирка с надписью “Plot expression” (график выражения).

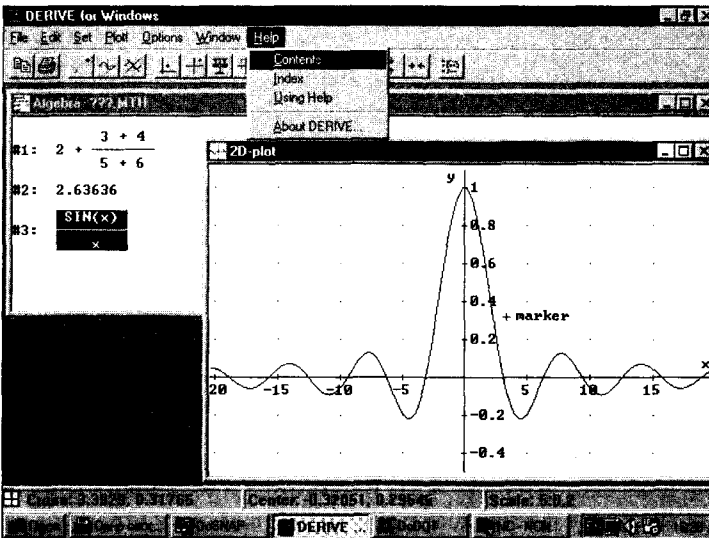
**Рис. 4.8**  
График функции –  
выделенного выражения  $\text{SIN}(x)/x$



А куда пропало окно с выражениями? Да никуда, просто оно заслонено окном вывода графика. Вы, вероятно, знаете, что любое окно в Windows 95 имеет в правом верхнем углу три характерные кнопочки управления: левая (кнопка Свернуть) сворачивает окно, средняя (кнопка Развернуть/Восстановить) устанавливает режим полностью или частично открытого окна, а правая закрывает окно. Будьте осторожны с правой кнопочкой: закрыв окно, вы теряете все размещенные в нем данные! Но активизировав среднюю кнопку, вы можете частично свернуть окно графики и обнаружить за ним окно математических выражений (см. рис. 4.9).

Оба окна можно перетаскивать мышью, менять их размеры и располагать в любом порядке; сверху всегда находится активное в текущий момент

**Рис. 4.9**  
Панель Derive  
с различными  
окнами



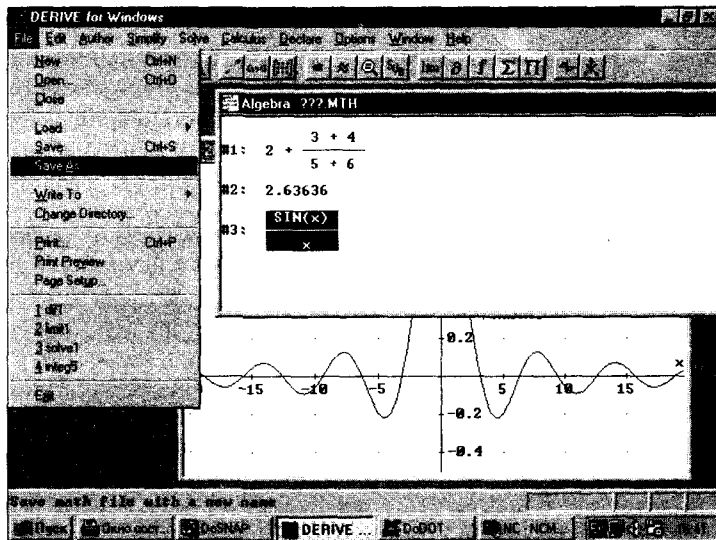
окно. Признаком активности является также темно-синяя верхняя строка заголовка окна. У неактивного окна эта строка серая. Для придания окну статуса активности достаточно “щелкнуть” курсором мыши по любому видимому участку окна, но только не по кнопкам управления. Под главным меню всегда располагается панель инструментов активного окна, в нашем случае — графического.

Итак, мы провели первый успешный эксперимент работы с Derive под Windows. Нетрудно заметить, это оказалось весьма необременительно. Особенно для пользователей, имеющих хотя бы некоторые навыки работы с приложениями под Windows. Однако для завершения знакомства осталось сделать последний шаг: записать введенные и вычисленные выражения на магнитный диск в виде файла, с тем чтобы в дальнейшем результаты наших экспериментов не пропали даром и ими можно было пользоваться для совершенствования навыков работы с системой и возможного применения.

#### 4.1.5. Сохранение сессии на диске

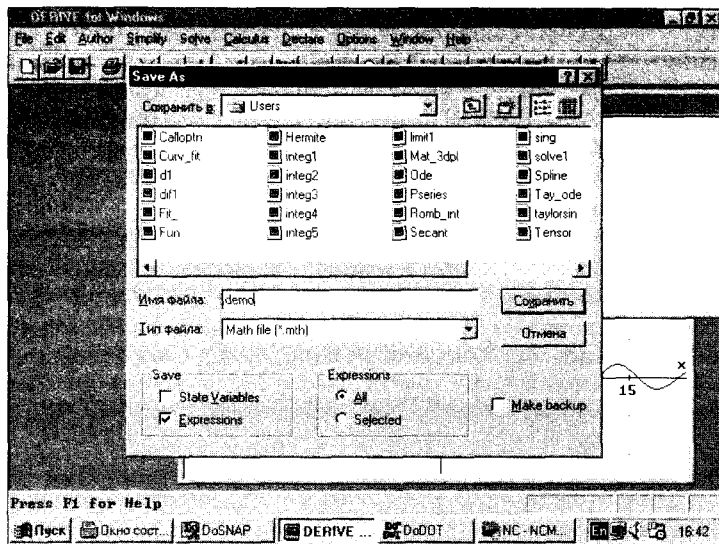
Для записи на диск выражений надо активизировать позицию File (Файл) главного меню. При этом возникает подменю, показанное на рис. 4.10. В этом подменю надо выбрать команду Save As... (Сохранить как...).

**Рис. 4.10**  
Подменю  
позиции  
File главного  
меню



Прежде чем описывать, что произойдет дальше, надо отметить, что окно с выражениями вначале не имеет имени, точнее говоря, имеет имя ??? .МТН. Здесь ??? — это имя окна, а .МТН — расширение файла. Команда Save As... позволяет записать файл под любым выбранным вами именем, а более быстрая команда Save под уже установленным именем. Если использована команда Save As... (наверняка вы захотите дать своему первенцу более содержательное имя, чем ???), в результате ее исполнения появится диалоговое окно записи, показанное на рис. 4.11. В нем можно установить имя файла и его тип, после чего для записи достаточно нажать кнопку ОК.

**Рис. 4.11**  
Диалоговое  
окно записи  
файла с задан-  
ным именем



Возможно, вы будете шокированы тем странным обстоятельством, что в этом окне часть надписей дана на русском языке. И это в то время, когда вас только что предупредили, что при работе с Derive о нашем языке надо забыть.

Увы, дело не в Derive, а в операционной системе Windows 95. На компьютере автора (надеюсь, и на вашем) стоит русифицированная версия этой операционной системы. Ее окна используются некоторыми приложениями (в том числе Derive), а потому изредка в них возникают надписи на русском языке. Как жаль, что содержательная часть мощнейшей системы помощи Derive (кстати, ее подменю видно на рис. 4.9) остается на английском языке.

## 4.2. Основы пользовательского интерфейса Derive под Windows

Теперь пора более подробно ознакомиться с пользовательским интерфейсом Derive под Windows. Верхнюю часть панели Derive занимает строка заголовка. На ней размещена надпись с именем системы. В правой части строки расположены уже упомянутые кнопки управления окном: Свертывания, Развертывания/Восстановления и Закрытия окна.

### 4.2.1. Главное меню

Дерево команд управления системой в новой версии существенно изменено, поэтому с ним надо познакомиться отдельно. Оно представлено главным меню и ниспадающими из его активных позиций подменю. Главное меню системы занимает вторую строку вверху окна (в версиях под MS-DOS главное меню располагалось сверху). Оно содержит следующие команды:

- File** — работа с файлами и принтером;
- Edit** — редактирование выражений;
- Author** — ввод математических выражений для их последующего преобразования;

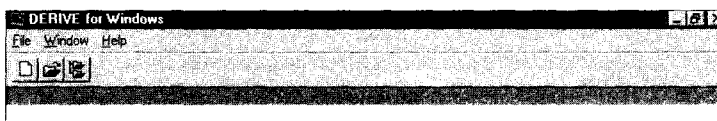


- Solve** — решение математических уравнений;
- Calculus** — вычисление производных, интегралов, пределов функций, сумм, произведений и разложение в ряд Тейлора;
- Declare** — декларация (задание) функций и переменных;
- Options** — задание различных опций (установок) системы;
- Window** — открытие новых окон и работа с окнами;
- Help** — включение справочной системы.

Главное меню, являющееся центром управления системы, дает доступ ко всем ее командам. Она может модифицироваться в зависимости от текущего состояния системы.

Например, если у Derive закрыты все окна, главное меню модифицируется и приобретает вид, показанный на рис. 4.12.

**Рис. 4.12**  
Вид верхней части переднего панно при закрытых окнах



Активизация позиций главного меню обеспечивается указанием на них курсором мыши и нажатием ее левой клавиши, перемещением выделения с помощью клавиш горизонтального перемещения курсора (для выбора команды в подменю используются клавиши вертикального перемещения курсора) и использованием “горячих” клавиш сочетаний — Alt и клавиши с подчеркнутой снизу буквой в названии команды. Эти способы управления применяются и для исполнения других команд, “горячие” клавиши для них указаны после имени команд.

Главное меню не является неизменным. Его вид изменяется в зависимости от состояния системы. Например, графические окна содержат свое главное меню. Некоторые команды в определенных ситуациях недоступны. Их надписи при этом даются затененным шрифтом, а порой соответствующие позиции меню попросту отсутствуют.

## 4.2.2. Панель инструментов и кнопки быстрого управления

В третьей сверху строке расположена панель инструментов. На ней размещен ряд кнопок быстрого управления системой. Каждая из них вводит одну из достаточно часто применяемых команд. Набор кнопок зависит от того, какое окно в настоящее время активно. Приведем назначения кнопок для окон с математическими выражениями. Они разбиты на семь групп (слева направо).

### Команды для работы с файлами:

- New** — открытие нового окна;
- Open** — вывод окна загрузки файла;
- Save** — запись файла под текущим именем.

### Команда печати содержимого окна с выражениями:

- Print** — печать содержимого окна с выражениями.

### Команды редактирования:

- Remove** — удаление строки с выделенным выражением;
- Unremove** — восстановление последней удаленной строки с выражением;
- Renumber** — нумерация строк в строгом порядке (1, 2, 3 и т. д.).

**Команды ввода:**

- Author expression** — ввод математических выражений;  
**Author vector** — задание векторов заданного размера;  
**Author matrix** — задание матриц заданного размера.

**Команды вычислений:**

- Simplify** — упрощение выражений;  
**Approximate** — вычисление выражений в числовой форме;  
**Solve** — решение уравнений и неравенств (и систем с ними);  
**Substitute for variable** — подстановки для переменных.

**Команды специальных вычислений:**

- Calculate limit** — вычисление пределов функций;  
**Calculate derivative** — вычисление производных функций;  
**Calculate integral** — вычисление неопределенных и определенных интегралов;  
**Calculate sum** — вычисление суммы рядов;  
**Calculate product** — вычисление произведения рядов.

**Команды вывода графических окон:**

- 2D-plot window (Ctrl + 2)** — вывод окна двумерной графики;  
**3D-plot window (Ctrl + 3)** — вывод окна трехмерной графики.

Применение кнопок панели инструментов дает удобный, особенно для новичков, способ управления системой. Но не стоит забывать и о других способах — управлении с помощью главного меню и “горячих” клавиш. Кнопки быстрого управления вводят только самые распространенные и часто встречающиеся команды.

### 4.2.3. Строка состояния системы

В нижней части панели расположена строка состояния. Ее назначение — оперативный контроль за работой системы. В этой строке выводятся текстовые комментарии — аннотации о статусе выделенных выражений. На них надо обратить особое внимание, поскольку по этим комментариям можно судить о том, какие действия выполнялись над выражениями и из каких других выражений они получены. Без этих комментариев документ *Derive* выглядит, как просто набор пронумерованных математических выражений, назначение которых трудно определить. Строка также дает пояснения о выполняемых командах и сообщает некоторую другую служебную информацию, например о времени (в секундах) выполнения расчетов и правильности их завершения, о параметрах графиков для графических окон и пр.

## 4.3. Работа с файлами (File)

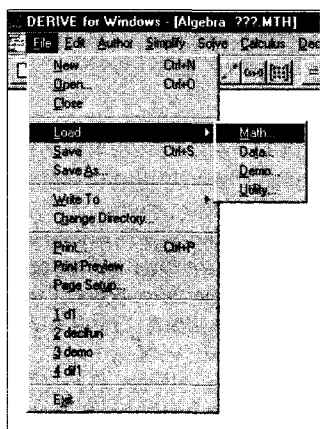
Позиция **File** главного меню содержит подменю со следующими командами для работы с файловой системой и печатающим устройством — принтером:

- New (Ctrl + N)** — создание нового окна выражений;  
**Open... (Ctrl + O)** — вызов диалогового окна загрузки файла документа;

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| Close               | — закрытие активного окна;                               |
| Load                | — вызов подменю загрузки различных файлов;               |
| Save (Ctrl + S)     | — вызов диалогового окна записи файла с заданным именем; |
| Save As...          | — запись документа с текущим именем;                     |
| Write To            | — запись в формате других языков программирования;       |
| Change Directory... | — изменение текущего каталога с файлами;                 |
| Print... (Ctrl + P) | — вызов диалогового окна начала печати;                  |
| Print Preview       | — предварительный просмотр страниц печати;               |
| Page Setup...       | — установка параметров страницы печати;                  |
| Exit                | — завершение работы с системой Derive и выход из нее.    |

Вид подменю позиции File показан на рис. 4.13. Обратите внимание, что после названия некоторых команд стоит маленький темный треугольник с вершиной, направленной из окна. Такие команды (например, Load) создают дополнительное подменю с набором различных команд.

**Рис. 4.13**  
Подменю  
позиции File  
главного меню



Рассмотрим назначение команд, относящихся к этой позиции главного меню. Большинство из них — типовые для приложений под Windows.

### 4.3.1. Создание нового окна выражений (New)

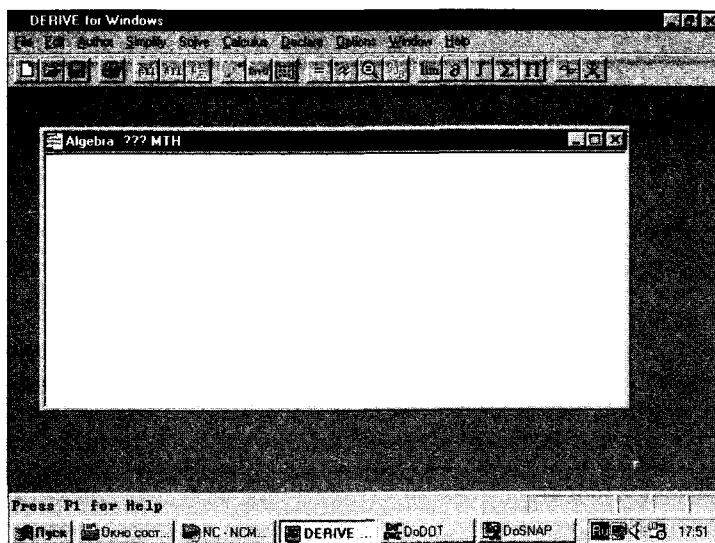
Для создания нового окна выражений служит команда New (новое). Эта команда обычно используется для подготовки нового документа. Она уничтожает все ранее открытые окна, создает новое чистое окно для ввода выражений (рис. 4.14) с именем ??? МТН.

Это окно можно перемещать и изменять его размеры. Нажав среднюю кнопку в конце строки заголовка окна, можно полностью развернуть окно; оно при этом займет весь экран (за исключением обязательных элементов интерфейса).

### 4.3.2. Вызов диалогового окна загрузки файла документа (Open...)

Команда Open вызывает появление на экране стандартного диалогового окна системы Windows, с помощью которого можно путешествовать по файловой системе компьютера. По умолчанию окно открывает доступ к папке — каталогу, который

**Рис. 4.14**  
Переднее  
панно системы  
с открытым  
чистым окном



является текущим. Но с помощью окна можно войти в любую папку в любом каталоге, расположенном на любом разделе жесткого диска или на гибких дисках. Вид окна рассмотрим чуть позже при описании команды Load. Заметим лишь, что окно называется Open и служит для загрузки файлов с расширением .MTH.

### 4.3.3. Закрытие активного окна (Close)

Как уже отмечалось, одновременно может быть активным только одно окно. Если это окно стало ненужным, его можно закрыть с помощью команды Close (закрыть). Тогда закрытое окно исчезает с экрана, но другие окна (если они были) остаются на экране. Если содержимое закрываемого окна не было сохранено на диске, система предупредит об этом в небольшом окошке и предложит сохранить (или не сохранить) содержимое окна в виде файла.

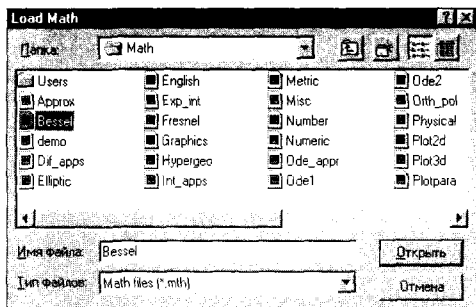
### 4.3.4. Работа с подменю загрузки различных файлов (Load)

Файлы в Derive могут быть различного типа: файлы документов, библиотек, демонстраций и т. д. Поэтому команда Load порождает подменю с командами:  
**Math...** — вызов диалогового окна загрузки документов с выражениями;  
**Data...** — вызов диалогового окна загрузки файла с данными;  
**Demo...** — вызов диалогового окна загрузки демонстрационных файлов;  
**Utility...** — вызов диалогового окна загрузки утилит.

Команда Math аналогична Open и обеспечивает вызов диалогового окна для загрузки файлов с расширением .MTH (см. рис. 4.15). Малая панель вверху с переключателем позволяет искать нужную папку, содержимое которой видно на большой панели в центре экрана. На панелях, расположенных ниже, можно установить имя файла и указать его тип. Переключатель тут имеет две позиции: выбор файлов с расширением .MTH и файлов \*.\* с любым расширением (напоминаем, что символы \* означают любое имя до восьми символов и любое расширение до трех символов).

Рис. 4.15

Диалоговое окно загрузки файлов с математическими выражениями



После загрузки файла с расширением .MTH его содержимое появляется в окне выражений и располагается в строго по порядку пронумерованных строках. Заметим, что в самих файлах строки с выражениями номеров не содержат и разделяются пустыми строками.

На рис. 4.16 показан вид экрана после загрузки одного из файлов, содержащих определения функций Бесселя. Эти определения видны на экране, причем все они после загрузки становятся действующими, т. е. новые введенные функции можно использовать для расчетов.

Рис. 4.16

Вид экрана при загруженном файле с определениями функций Бесселя

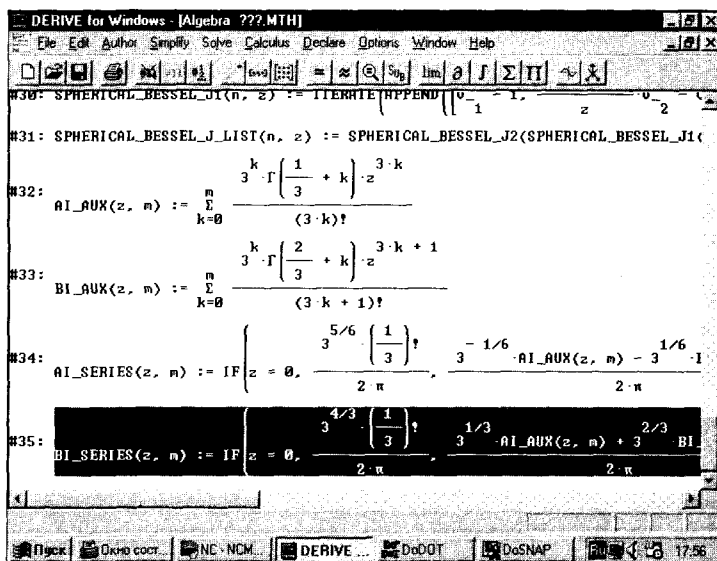


Рис. 4.17 подтверждает эту возможность — на нем задано вычисление в диалоговом окне Author Expression значения функции  $AI\_AUX(1, 1)$ . Что это такое, сейчас совершенно неважно. Главное, что такая возможность есть и она указывает на способность системы Derive к расширению как путем использования готовых утилит с новыми функциями, так и благодаря применению утилит, готовящихся самим пользователем.

На рис. 4.18 показан результат вычисления указанной функции. Как трудно заметить, он расположился внизу, в конце введенного документа с утилитами для вычисления функций Бесселя.

Рис. 4.17

Подготовка  
к вычислению  
функции, задан-  
ной в загружен-  
ном расширении

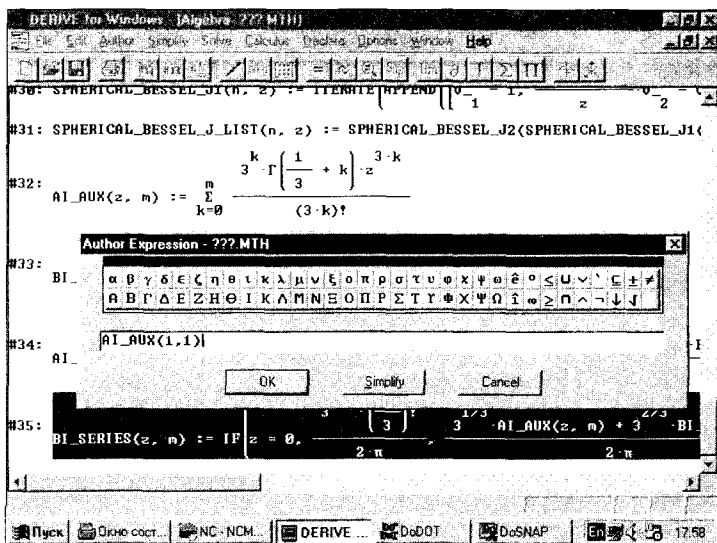
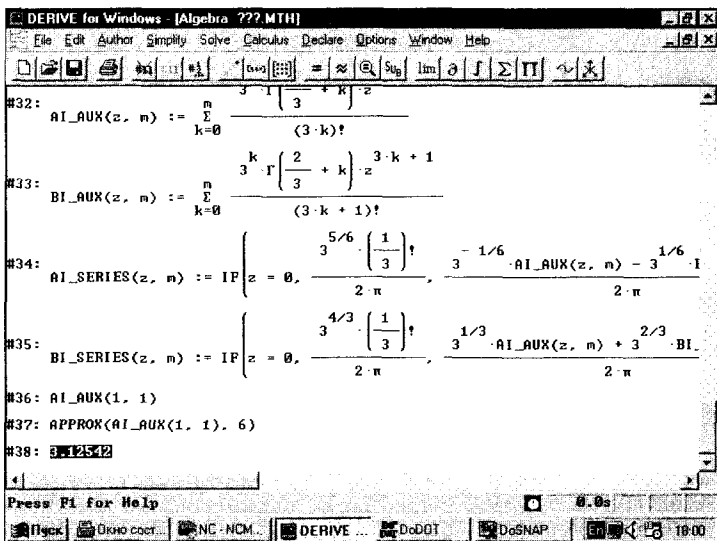


Рис. 4.18

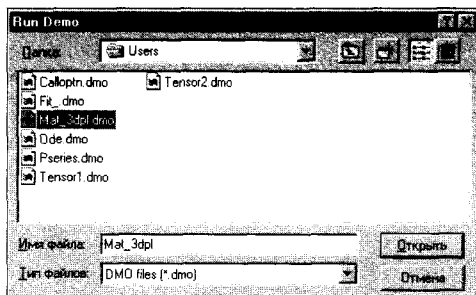
Пример  
вычисления  
одной из функ-  
ций, заданных  
в документе



Файлы данных имеют расширение .DAT и являются файлами текстового формата, но содержащими числа — от одного числа до двумерного массива (матрицы) с размером до  $100 \times 100$  элементов, представленных в общепринятом виде, т. е. в виде целых чисел или чисел с плавающей точкой. Благодаря возможности ввода таких файлов Derive позволяет обрабатывать данные, получаемые от внешних источников, например измерительных приборов.

В Derive под Windows существенно улучшен просмотр демонстрационных файлов. Напомним, что это файлы с расширением .DMO. Их отличительная черта — возможность построчного просмотра с выводом результатов вычислений по ходу просмотра. Для загрузки таких файлов используется коман-

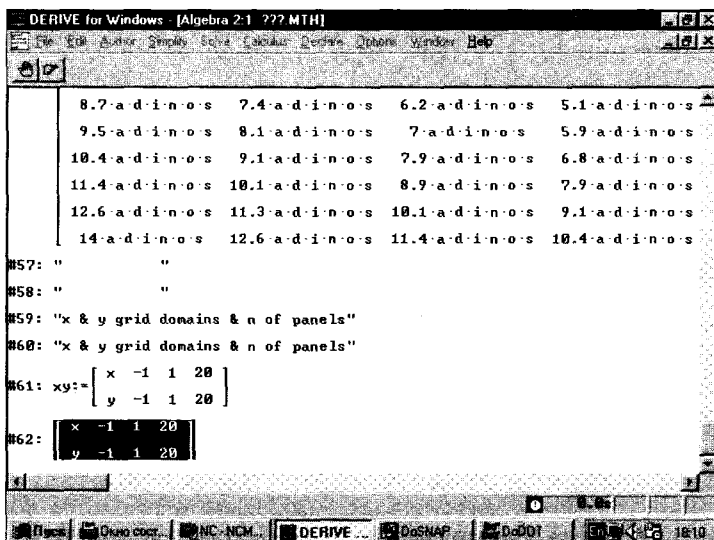
**Рис. 4.19**  
Диалоговое  
окно загрузки  
демонстрацион-  
ных файлов



да Demo. Она выводит диалоговое окно для поиска и загрузки демонстрационных файлов (см. рис. 4.19).

Это окно практически аналогично окну Load или Open и позволяет путешествовать по файловой системе компьютера. На рис. 4.19 показан момент выбора файла Mat\_3dpl.dmo из папки (каталога) Users. Для загрузки выбранного файла достаточно нажать клавишу ОК. На рис. 4.20 показан вид экрана в режиме просмотра демонстрационного файла. Нетрудно заметить, что окно освобождается от второстепенных деталей и в левом верхнем углу появляются две кнопки в виде изображения ладони: одна (правая) указывает на пуск просмотра, а другая — на его остановку.

**Рис. 4.20**  
Derive под Win-  
dows в режиме  
просмотра де-  
монстрацион-  
ного файла

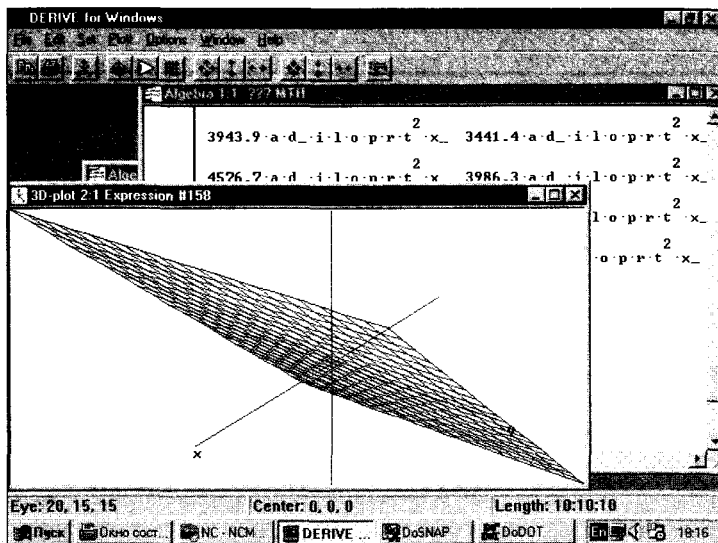


При каждом нажатии на кнопку просмотра выполняется очередная строка просматриваемого файла, и тут же появляется результат вычисления, разумеется, если в строке задано исполняемое выражение. Строки с текстовыми комментариями просто повторяются. По завершении просмотра законченного блока демонстрационного файла можно увидеть окончательный результат и, если это необходимо, построить график (рис. 4.21).

Еще один важный вид файлов — утилиты, входящие в библиотеки расширения. Фактически это те же файлы с расширением .MTH. Однако Derive

Рис. 4.21

Пример построения графика по данным демонстрационного файла



имеет специальную команду Utility их загрузки с помощью уже описанного диалогового окна загрузки файлов. Особенность загрузки файлов в последнем случае состоит в том, что она производится только в память компьютера и содержимое файлов не выводится на экран. Это дает два преимущества: экран не захламляется уже отлаженными и выверенными выражениями, а они загружаются существенно быстрее, чем файлы

#### 4.3.5. Запись документа с текущим именем (Save)

Если вы работали с ранее созданным файлом, значит, он имеет свое текущее имя. Команда Save служит для быстрой записи такого файла без изменения его имени. Она не создает никаких диалоговых окон. Рекомендуется время от времени пользоваться этой командой в ходе редактирования и модификации больших документов, которые нередко создаются достаточно долго. Тогда вы не рискуете потерять результаты своей работы из-за случайного выброса, отключения сети или иной аварии компьютера.

#### 4.3.6. Вызов диалогового окна записи файла с заданным именем (Save As...)

Команда Save As..., как уже отмечалось в начале главы, служит для записи файла на диск с изменением имени или с изменением места расположения файла. К примеру, ее следует использовать при копировании файла Derive с жесткого диска на гибкий или для создания копии файла с другим именем. Эта команда вызывает появление диалогового окна, которое было показано на рис. 4.11.

Хотя основные возможности этого окна (переименование файла и задание нового места его расположения) уже были обсуждены, сейчас надо отметить некоторые тонкости процесса записи. Внизу окна расположены некоторые важные опции записываемого файла, которые можно применить. Так, две позиции, State Variable и Expression, позволяют задать запись статуса пере-



менных и запись выражений. Для записи надо установить птичку в маленький прямоугольник.

Соответствующая группа опций Expression состоит из двух: All — запись в файл всех выражений и Selected — запись лишь выделенной цветом основы выражений. Включение опций задается установкой в кружки жирных точек.

Наконец, опция Make в Backup (обычно отключенная) позволяет использовать функцию резервного копирования файлов. Обычно эти опции используются по умолчанию, но знать их назначение весьма полезно.

Достаточно задавать только имя файлов (до 8 символов). Расширение Derive проставляет самостоятельно, так что вводить его не нужно, но можно. Не следует лишь вводить свои нетиповые расширения. Имена файлов должны нести смысловую нагрузку.

### 4.3.7. Запись в форматах языков программирования (Write To)

Позиция Write To главного меню содержит подменю со следующими командами:

**Basic file...** — запись выражений в формате файлов Бейсика;

**C file...** — запись выражений в формате файлов Си;

**Fortran file...** — запись выражений в формате файлов Фортрана;

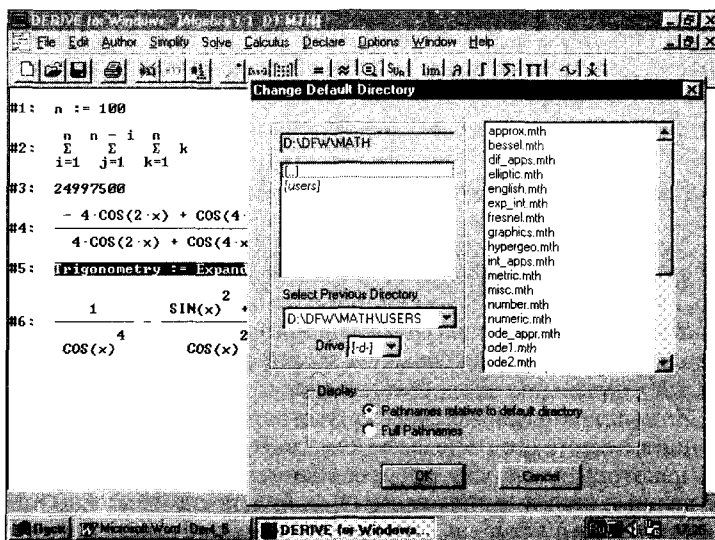
**Pascal file...** — запись выражений в формате файлов Паскаля.

Эти команды позволяют записывать выражения в формате файлов популярных языков программирования Бейсик, Си, Фортран и Паскаль. Они могут потребоваться при совместном использовании Derive и программ на этих языках. Указанные команды вызывают диалоговое окно, практически аналогичное окну Save As...

### 4.3.8. Изменение текущего каталога с файлами (Change Directory...)

При использовании команд загрузки и записи файлов Derive выводит диалоговые окна с установленным каталогом. Однако если вы хотите изменить этот каталог, достаточно воспользоваться командой Change Directory... (смена каталога). Она вызывает появление диалогового окна, показанного на рис. 4.22.

Рис. 4.22  
Диалоговое окно  
смены каталога



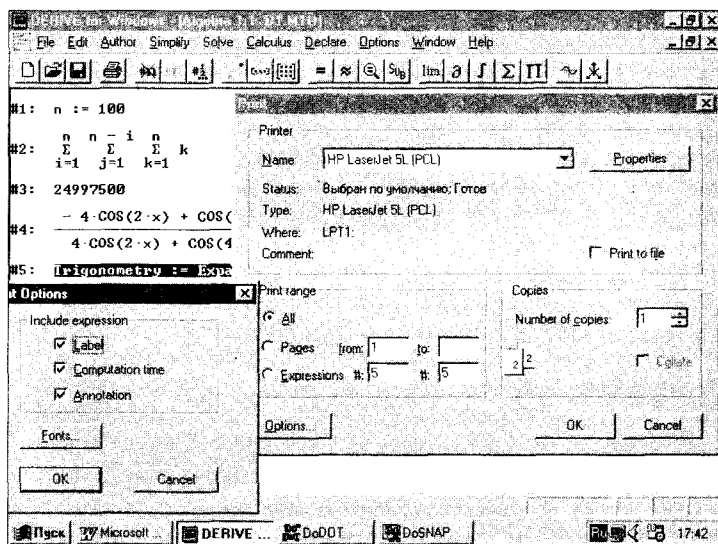
С помощью этого окна можно найти нужный каталог. В большом информационном поле окна находится список файлов, существующих в выбранном каталоге.

### 4.3.9. Вызов диалогового окна начала печати (Print...)

Документ или график в активном окне может быть распечатан на принтере, который установлен операционной системой. В отличие от версий Derive под MS-DOS версия под Windows не содержит своего драйвера печати и использует для нее средства Windows. Однако надо учитывать, что Derive применяет свои наборы шрифтов для вывода как на экран дисплея, так и на принтер.

Команда Print выводит на экран диалоговое окно, вид которого зависит от установленного принтера, а точнее, от драйвера, обеспечивающего работу принтера в среде Windows. На рис. 4.23 показан вид окна, характерного при работе с лазерным принтером Hewlett-Packard Laser Jet 5L.

**Рис. 4.23**  
Диалоговое окно  
начала печати



Чтобы начать распечатку, достаточно активизировать кнопку ОК, а чтобы отказаться от печати — кнопку Cancel. В этом окне можно установить некоторые параметры печати, в частности диапазон страниц, которые нужно напечатать, и число копий. Кнопка Properties (Свойства) позволяет вывести другое окно с установкой целого ряда параметров печати: размера страниц, ориентации и качества печати и т. д. Вид этого окна тоже может быть различным в зависимости от установленного принтера.

Кнопка Options (Опции) выводит на экран небольшое диалоговое окно опций печати. Оно показано на рис. 4.23 в нижнем левом углу. Это окно позволяет установить следующие опции при печати:

- Label** — задает печать меток (номеров строк);
- Computation time** — задает печать времени вычислений;
- Annotation** — печать аннотаций.

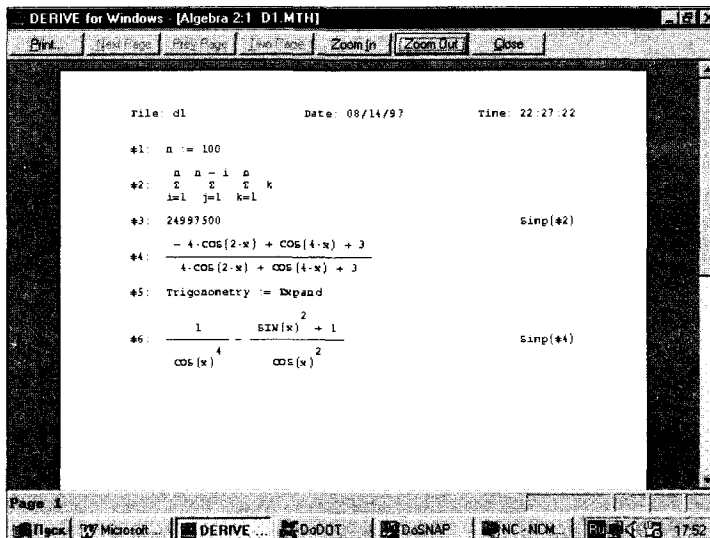
Кроме того, кнопка **Fonts...** позволяет заменить набор символов (фонтов) и при необходимости выполнить печать не только специальным шрифтом из набора шрифтов Derive, но и другими, установленными в операционную систему Windows.

### 4.3.10. Предварительный просмотр страниц печати (Print Preview)

В большинстве случаев, прежде чем начать распечатку документов или графиков, стоит посмотреть, как они будут располагаться на листах бумаги. Для этого служит команда **Print Review**. Она выводит на экран страницу с документом или с графиком, расположенными в том виде, в каком они будут напечатаны (см. рис. 4.24).

Окно просмотра имеет средства для изменения масштаба страницы в ходе ее просмотра и для переключения страниц. Кнопка **Print...** позволяет начать распечатку наблюдаемой страницы прямо из окна предварительного просмотра.

**Рис. 4.24**  
Панель Derive в режиме предварительного просмотра страницы



### 4.3.11. Установка параметров страницы печати (Page Setup...)

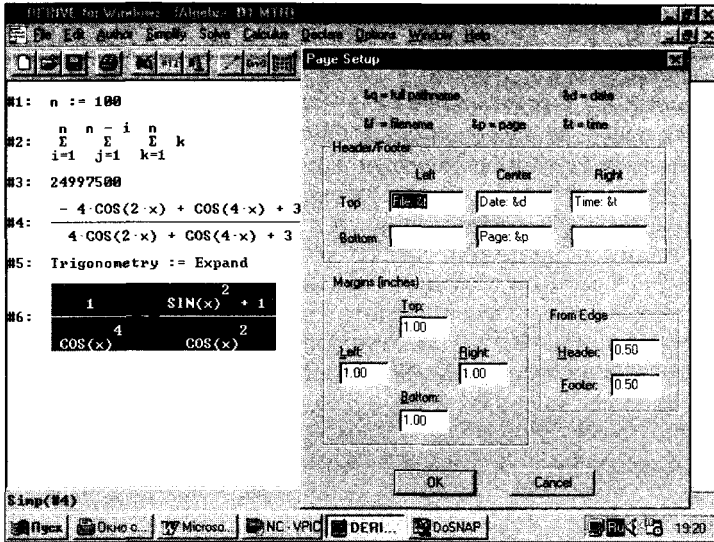
Если расположение объектов печати на странице вас не устраивает, следует использовать команду установки параметров страницы печати **Page Setup...** На рис. 4.25 показано диалоговое окно установки параметров страницы печати, которое появляется при исполнении команды.

Здесь можно установить места отображения имени файла, даты, времени и номера страницы. Кроме того, в панели **Margins** можно установить размеры полей сверху (**Top**), справа (**Right**), слева (**Left**) и снизу (**Bottom**). Размеры указываются в дюймах.

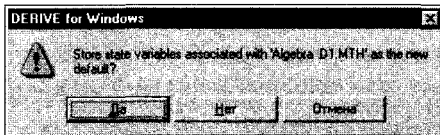
### 4.3.12. Завершение работы с системой Derive и выход из нее (Exit)

Для завершения работы с Derive служит команда **Exit** (**Выход**) в подменю позиции **File** главного меню. При исполнении этой команды появляется диалоговое окно, показанное на рис. 4.26.

**Рис. 4.25**  
Панно Derive  
с окном установки  
параметров  
страницы печати



**Рис. 4.26**  
Диалоговое окно  
завершения работы  
с Derive



В этом окне спрашивается, нужно ли перед завершением работы с Derive сохранить документ или значения переменных измененными. Три кнопки с надписями на русском языке (вот он, парадокс работы с русифицированной версией Windows 95) позволяют подтвердить запись — “Да”, отказаться от нее — “Нет” и отказаться от завершения работы вообще — “Отмена”.

## 4.4. Редактирование выражений и документов (Edit)

Основные операции и команды по вводу различных режимов редактирования сосредоточены в подменю позиции Edit (Редактор) главного меню. Она содержит несколько команд, разбитых на три группы. В первой группе сосредоточены команды вывода окон редактирования выражений, аннотаций и указания переходов к заданному выражению:

- Expression...** — вывод окна редактирования выделенного выражения;
- Annotation...** — вывод окна аннотации для выделенного выражения;
- Go to Expression...** — вывод окна с указанием номера выражения, к которому осуществляется переход.

Вторая группа команд служит для работы со строками документа:

- Remove... (Ctrl + R)** — стирание выделенного выражения (строки);
- Unremove... (Ctrl + U)** — восстановление последнего стертого выражения;
- Move... (Ctrl + M)** — перемещение выражения;
- Renumber** — перенумерация строки с выражениями.

Наконец, две команды последней группы служат для работы с буфером обмена, используемого в Windows:

**Copy Expressions (Ctrl + C)** — копирование выражения в буфер обмена;

**Mark and Copy... (Ctrl + Shift + M)** — выделение и копирование выражения.

Как известно, буфер обмена (Clipboard) в операционной системе Windows является специально организованной и динамически изменяемой областью памяти, в которую можно помещать различную информацию. В известной степени буфер интеллектualен, так как “понимает” различные форматы информации, например текстовый и графический в системе Derive. С помощью буфера возможен обмен такой информацией между приложениями под Windows.

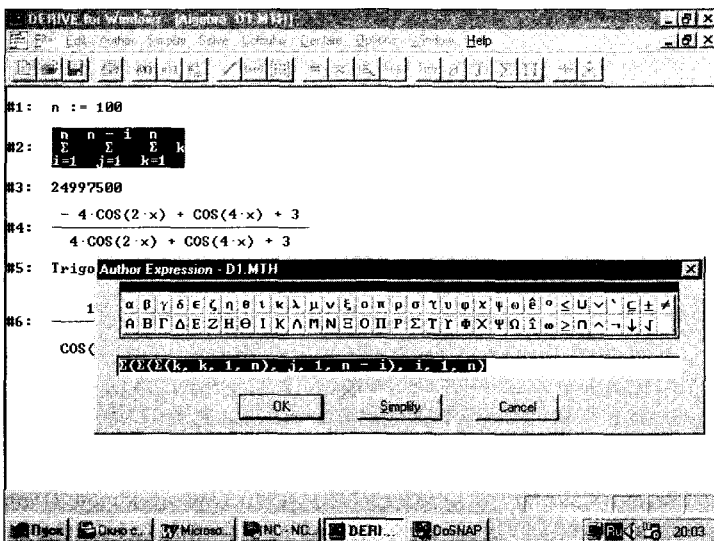
Не все команды этой позиции главного меню доступны в любое время. Например, если в окне выражений нет никакого выражения, большинство команд окажутся недоступными, поскольку они будут лишены смысла. Такие команды в подменю отображаются затененными надписями. Это правило действует в отношении команд и в других подменю, а также в отношении кнопок быстрого управления системой; они выглядят блеклыми, если недоступны.

#### 4.4.1. Вывод окна редактирования выделенного выражения (Expression...)

Команда Expression... обеспечивает вывод окна для набора или редактирования выделенного выражения. Это окно показано на рис. 4.27.

**Рис. 4.27**

Панно Derive с окном редактирования выделенного выражения



Если в окне выражений какое-либо выражение выделено цветом, то команда Expression... доступна и появляется указанное окно. Вновь отметим, что оно содержит наборную панель специальных символов и строку редактирования.

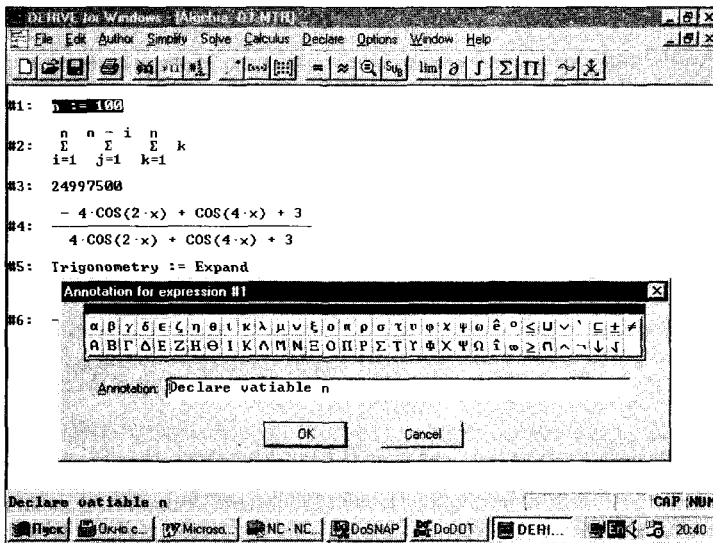
Редактирование происходит по правилам, принятым для приложений под Windows. Главное при этом то, что для перемещения текстового маркера используются клавиши перемещения курсора по вертикали. Клавиша Ins по-

звляет задавать режимы вставки символа и замещения. С помощью клавиши Shift и клавиш перемещения курсора по горизонтали можно осуществлять выделение любой части выражения. Выделенное выражение можно убрать в буфер, нажав клавиши Shift и Del, а хранящийся в буфере фрагмент выражения вставить в любое место редактируемого выражения, установив на это место текстовый маркер и нажав клавиши Shift и Ins.

#### 4.4.2. Ввод аннотации для выделенного выражения (Annotation...)

Каждое выражение может снабжаться аннотацией, т. е. небольшим текстовым комментарием. Аннотация отображается в строке состояния, если выражение выделено. Для ввода аннотации служит команда Annotation..., которая выводит на экран окно ввода и редактирования аннотации (см. рис. 4.28).

**Рис. 4.28**  
Окно ввода аннотации к выделенному выражению

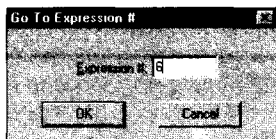


Окно ввода аннотации подобно окну редактирования выражения, но на нем нет кнопки Simplify. Правила ввода и редактирования аннотаций те же, что и для выражений. Если выделенное выражение ранее было снабжено аннотацией, то она окажется в строке вывода аннотаций; в этом случае аннотацию можно редактировать. Если аннотации не было, строка ввода окажется пустой, и можно вводить аннотацию. На рис. 4.28 показана аннотация о декларировании переменной  $n$  как в строке ввода, так и в строке состояния. Ввод аннотаций (увы, только англоязычных) значительно упрощает разбор сложных документов, в которых используются выражения в разных строках и в различной последовательности.

#### 4.4.3. Переход к заданному выражению (Go to Expression...)

При работе со сложными документами и их редактировании нередко приходится обращаться к заданной строке. Для этого используется команда Go to Expression... Она вызывает появление в окне выражений еще одного небольшого окна для указания номера выражения, к которому задается переход (см. рис. 4.29).

**Рис. 4.29**  
Окно ввода номера выражения, к которому задается переход

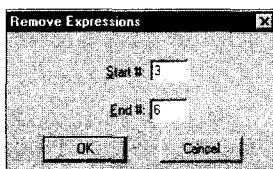


Если подтвердить ввод номера выражения активизацией кнопки ОК, произойдет переход к этому выражению и его выделение. Для отказа от перехода используется кнопка Cancel.

#### 4.4.4. Стирание строк с выражениями (Remove...)

Окно выражений при практической работе с Derive довольно быстро засоряется ставшими ненужными или ошибочными выражениями и результатами их вычислений. Поэтому часто возникает необходимость стирания того или иного выражения. Для стирания строк с выражениями служит команда Remove. Она вызывает появление окна (рис. 4.30), в котором можно указать диапазон стираемых строк, например от 3-й до 6-й.

**Рис. 4.30**  
Окно задания номеров стираемых строк



Если в окне выражений какое-либо выражение выделено, оно окажется в обеих позициях окна задания номеров стираемых строк. Вообще, если надо стереть какую-то одну строку, надо задать ее номер в позициях Start# и End#.

#### 4.4.5. Восстановление стертых строк (Unremove...)

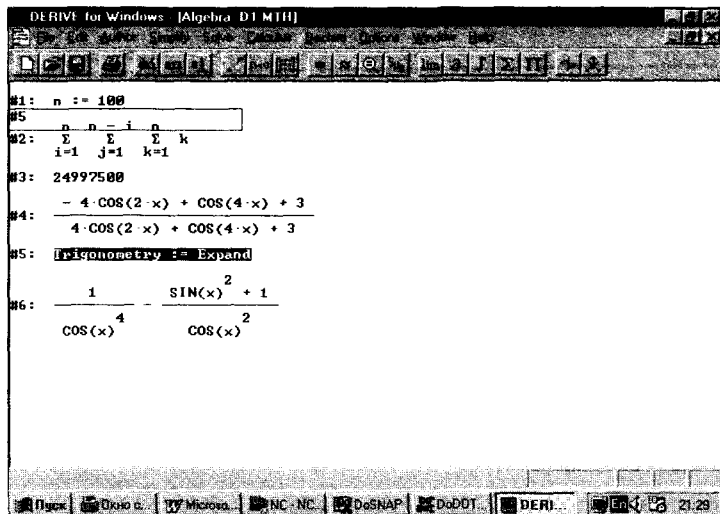
Нередко случается, что сразу после стирания одной или ряда строк обнаруживается, что стерта нужная строка или даже несколько нужных строк. Тут нет ничего страшного — команда Unremove... позволяет ликвидировать последнюю процедуру стирания и восстановить стертые строки.

#### 4.4.6. Перемещение выражений (Move...)

В ходе редактирования сложных документов часто возникает необходимость в перемещении строки или группы строк из одного места документа в другое. Прежде чем обсуждать, как производится такое перемещение с помощью команды Move..., отметим, что есть возможность перемещения выражения с помощью мыши. Для этого достаточно подвести курсор мыши к выделенным выражениям и нажать правую клавишу мыши. Удерживая ее нажатой, перемещайте мышь вверх или вниз. Вы увидите, что макет выделенного выражения (или блока выражений) в виде красной рамки с номером будет перемещаться вслед за движением мыши (см. рис. 4.31).

Как только вы установите макет выражения в новое место, надо отжать правую клавишу мыши. Выражение окажется на новом месте (со своим номером). Перенумерации строк при перемещениях отдельных выражений не происходит.

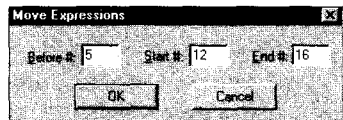
**Рис. 4.31**  
Перемещение  
строки с выраже-  
нием мышью



Если вы заранее знаете, куда надо переместить блок из нескольких строк, целесообразно использовать команду Move... Она вызывает появление диалогового окна, показанного на рис. 4.32.

Это окно имеет три панели для ввода:

**Рис. 4.32**  
Окно  
перемещения  
строк



- номера строки Before#, перед которой размещается блок строк;
- номера начальной строки блока Start#;
- номера последней строки блока End#.

Например, если вы хотите перед строкой 5 поместить блок из нескольких строк от 12 до 16, установите:

**Before#: 5    Start#: 12    End#: 16**

Для перемещения одной строки надо указать ее номер как в позиции Start#, так и в позиции End#. При использовании команды Move перенумерация строк также не происходит.

#### 4.4.7. Перенумерация строк (Renumber)

Работать со строками, нумерация которых оказывается сбитой перемещениями и стиранием части строк, не очень удобно. Если вы это почувствовали, можете командой Renumber произвести нумерацию строк в строго последовательном порядке. Derive автоматически восстановит связи между выражениями, если они оказались нарушенными из-за изменения нумерации строк. Эта команда очень напоминает команду изменения нумерации строк в древнейших версиях языка Бейсик и лишний раз подтверждает мнение о том, что Derive — это математически ориентированный Бейсик, разумеется, в новейшем обличии.

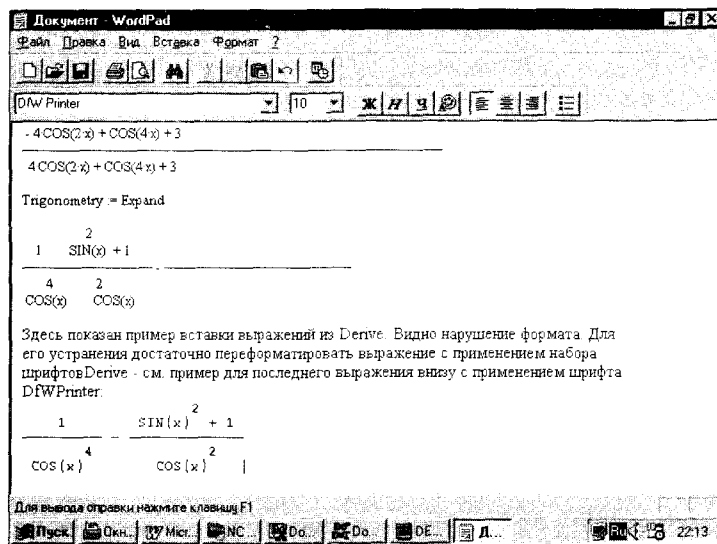


#### 4.4.8. Копирование выражений в буфер обмена (Copy Expressions)

Любую выделенную строку с выражением или блок из нескольких строк можно поместить в буфер обмена операционной системы Windows, используя команду Copy Expression. Номера строк при этом не копируются. Тут возможна передача содержимого буфера в другое приложение под Windows при условии, что в позиции Edit его главного меню имеется команда Paste (Вставить). Содержимое буфера при этом помещается в место, обозначенное курсором мыши.

Интересно, что с помощью этой команды можно копировать и рисунки. Но не все графические данные из буфера копируются без искажений. Например, в текстовый редактор Word 95 (Word 7.0) копирование рисунков Derive происходит с заметными искажениями. А при копировании выражений сохраняется текстовый формат и текущий шрифт. Для устранения искажений приходится переформатировать выражения с использованием набора шрифтов Derive (см. рис. 4.33).

**Рис. 4.33**  
Пример передачи содержимого буфера в текстовый редактор WordPad и переформатирования выражений



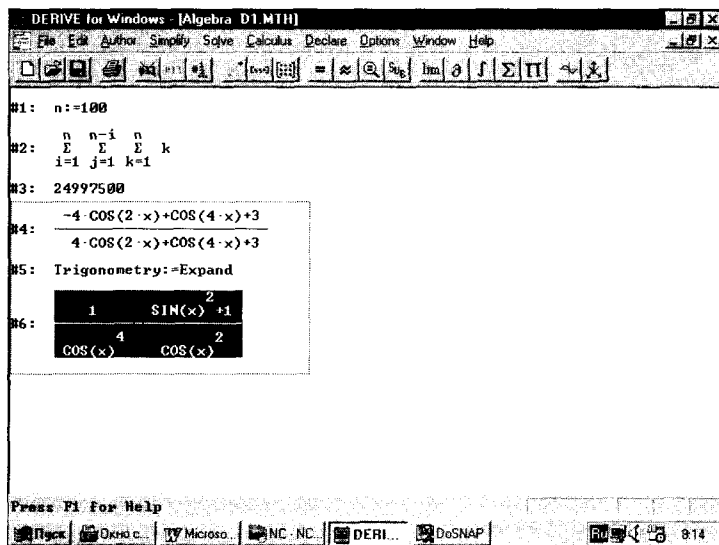
Механизм обмена информацией через буфер открывает возможность создания рабочих мест математика (или школьника и студента). Они станут применять Derive для сугубо математических действий, а подходящий текстовый редактор (Word 7.0 или WordPad) — для окончательной подготовки документов, включающих в себя описание на любом языке, в том числе и русском. Позже мы покажем, что так же можно копировать в буфер графики, создаваемые Derive, и вставлять их в тексты редакторов класса Word.

#### 4.4.9. Выделение и копирование выражения (Mark and Copy...)

Команда Mark and Copy... похожа на предыдущую, но обладает заметно большими возможностями. В частности, она позволяет мышью выделять любой фрагмент изображения в окнах выражений или графиков. Для это-

го надо исполнить команду — курсор мыши приобретет вид тонкого креста. Затем надо, держа нажатой левую клавишу мыши, перемещать ее, при этом появится расширяющийся прямоугольник из пунктирных красных линий (см. рис. 4.34).

**Рис. 4.34**  
Выделение с помощью команды Mark and Copy...



Этот расширяемый прямоугольник позволяет выделить любое изображение и, после отжатия левой клавиши мыши, поместить его в буфер обмена. Если какое-то из выделенных выражений выделено еще и цветом, это выделение сохраняется в копии выделенного участка, находящейся в буфере. В общем, этот вариант копирования сохраняет графическую копию любого изображения в окнах Derive.

## 4.5. Создание рабочего места математика на базе Derive и текстового процессора класса Word

Сама по себе Derive обладает весьма ограниченными возможностями по описанию сессии работы с ней. Как уже отмечалось, в ее документах возможны лишь текстовые комментарии в кавычках и аннотации к строкам с выражениями. Этого недостаточно для полноценного описания документов, например, при подготовке курсовой или дипломной работы, отчета о научном исследовании или диссертации.

Сейчас разработчики программных средств уделяют большое внимание созданию автоматизированных рабочих мест математиков для школьников и студентов, инженеров и ученых. Назначение таких рабочих мест заключается в подготовке высококачественных документов, в которых используются любые шрифты и выделения в сочетании с математическими формулами и графиками.

Такие рабочие места создаются при объединении популярных текстовых процессоров с какой-либо универсальной математической системой. Например, рабочее место MathOffice for Word 6.0 создано объединением текстового

процессора Word 6.0 с математической системой Maple. Другое рабочее место, WorkPlace, объединяет текстовый редактор Scientific Word фирмы TCI Software Research с системой Maple и позволяет готовить документы в формате LATEX.

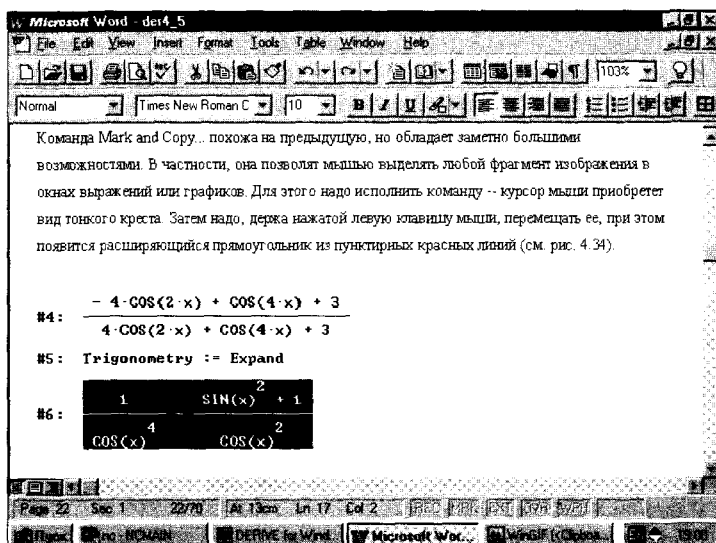
Покажем, что подобное рабочее место вполне можно создать на базе математической системы Derive и популярных текстовых редакторов Word 95 (Word 7.0) и WordPad (входит в Windows 95). Тут достаточно запустить Derive и выбранный текстовый редактор (процессор) одновременно. Derive при этом используется для проведения математических вычислений и преобразований и получения их результатов в символьной, численной или графической форме. Текстовый процессор служит для создания высококачественных описаний (в том числе на русском языке) в наиболее распространенных форматах с включением в них импортируемых данных из Derive в форме текстов документов и рисунков.

Все это достаточно просто при работе одновременно с Derive и текстовым редактором. Вы можете выделить какие-либо выражения командой Mark and Copy и поместить их в буфер обмена. Затем переключитесь в текстовый процессор, активизировав его бирку в панели приложений Windows 95 (она находится в самой нижней строке панно Windows 95) или выбрав знак редактора переключателем приложений — клавиши Alt и Tab.

Затем содержимое буфера можно поместить в окно текстового редактора, например Word 95 (см. рис. 4.35), используя команду Paste в подменю позиции File главного меню редактора.

Рис. 4.35

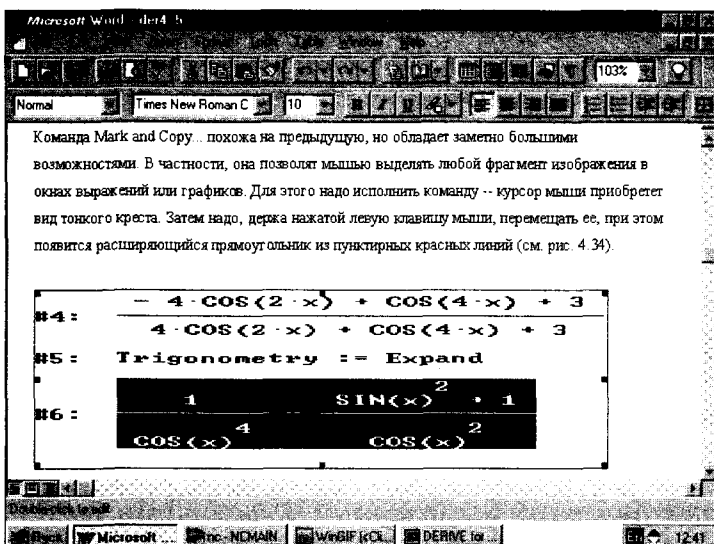
Пример вставки в Word 95 изображения, выделенного на рис. 4.34



Вставленное выражение или рисунок становятся графическим объектом в том приложении, которое его импортировало. Такой объект можно выделить рамкой с метками и растягивать в различных направлениях курсором мыши. На рис. 4.36 показано растягивание по горизонтали импортируемого из Derive в Word 95 изображения. Нетрудно заметить отсутствие здесь каких-либо искажений. Искажения отсутствуют и при импорте графиков из графических окон Derive.

Рис. 4.36

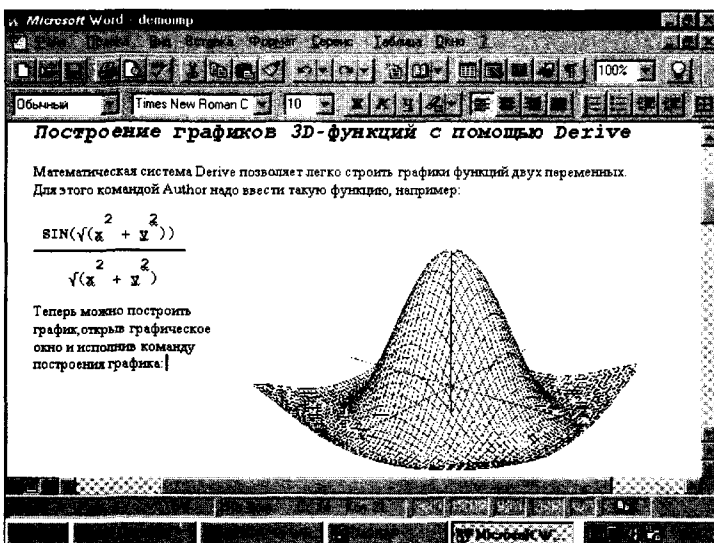
Пример растягивания импортируемого изображения в Word 95



Таким образом, используя возможности помещения данных Derive в буфер обмена, можно перенести эти данные в документы текстового процессора и готовить объединенные документы с применением всех возможностей последнего. На рис. 4.37 показан документ Word 95, в который включено выражение Derive в текстовом формате и график трехмерной функции. Они получены в результате использования команд Copy Expression и Mark and Copy у Derive и Плавка Вставить у редактора Word 95.

Рис. 4.37

Пример комбинированного документа в окне редактора Word 95



Отметим, что изображения рисунков в исходном масштабе в Word 95 воспроизводятся без искажений. Но при сильном сжатии рисунков (как на рис. 4.37) искажения все же наблюдаются. Это недостаток текстового процес-

сора Word 95. Как уже отмечалось, более простой текстовый редактор WordPAD в подобных случаях дает меньшие искажения.

## 4.6. Ввод данных (Author)

Для выполнения вычислений нужно в первую очередь ввести необходимые для этого данные, прежде всего математические выражения и данные векторного и матричного типов. Позиция главного меню Author выводит подменю, необходимое для осуществления этих операций. Оно содержит следующие команды:

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

Рассмотрим их более подробно.

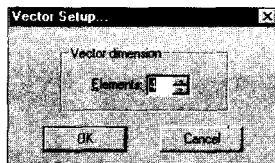
### 4.6.1. Ввод математических выражений (Expression...)

Команда Author Expression служит для ввода авторских выражений, над которыми выполняются последующие преобразования. Как уже отмечалось, эта команда вызывает появление окна ввода (рис. 4.3), с помощью которого в режиме строчного редактирования можно вводить различные математические выражения — как численные, так и в виде математических формул. Специальные математические знаки можно вводить, используя панель с их набором. С ее помощью вводятся и переменные, имена которых задаются греческими буквами.

### 4.6.2. Задание вектора определенных размеров (Vector...)

Вектор — массив из заданного количества математических выражений. Для задания вектора используется команда Vector..., которая вызывает появление окна для ввода размера вектора (см. рис. 4.38). Размер вектора можно ввести как непосредственно, так и с помощью счетчика справа от окошка ввода.

**Рис. 4.38**  
Окно ввода  
размера вектора



После ввода размера вектора появляется окно для ввода значений элементов вектора (см. рис. 4.39). Если задан ввод вектора большого размера (а он может быть от 1 до 100), справа от строк ввода появляется полоса вертикальной прокрутки.

Элементы векторов могут быть числами, именами переменных или символьными выражениями.

На рис. 4.40 приведены примеры векторных операций при работе с системой Derive под Windows. В основном здесь подобраны примеры, наглядно иллюстрирующие выполнение символьных операций с элементами векторов:

- вычисление факториала для элементов вектора — строки 1, 2;
- символьное вычисление трех производных функции #3 — строки 4, 5;
- создание переменной вектора из четырех производных функции  $x^n$  — строки 6, 7;
- вывод символьных значений переменной-вектора — строки 7, 8;
- интегрирование переменной-вектора — строки 9, 10.

Рис. 4.39

Панель ввода  
элементов  
вектора

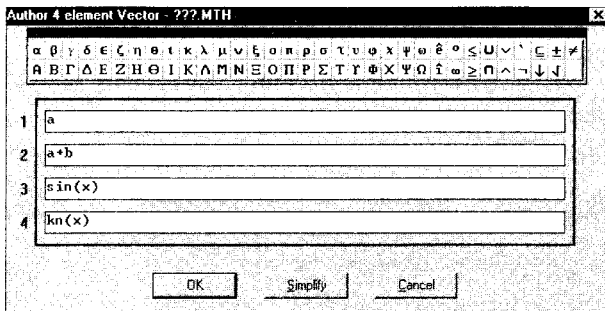
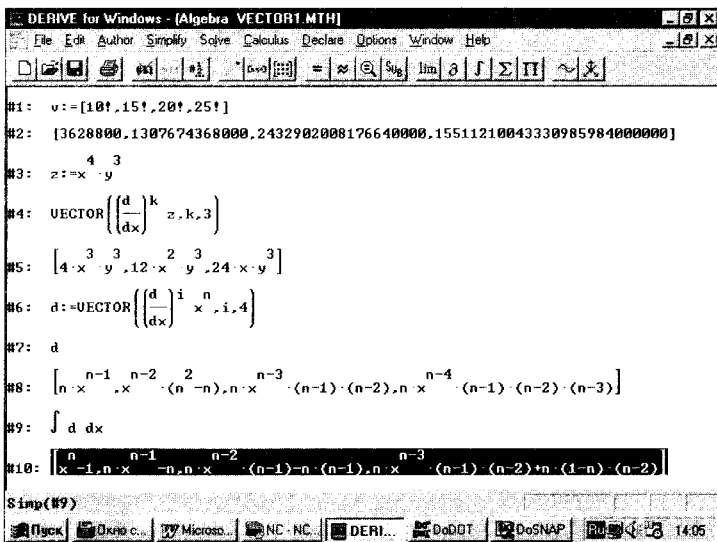


Рис. 4.40

Примеры работы  
с векторами

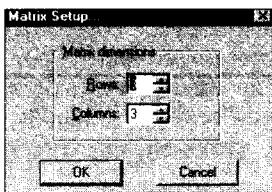


Операции с векторами были рассмотрены также в главах 2 и 3. Следует обратить внимание на то, что понятие вектора в Derive существенно расширено. Правильнее было бы говорить о списках или листах, как это принято в наиболее мощных пакетах символьной математики — Mathematica 2/3 или Maple V.

### 4.6.3. Определение матрицы заданных размеров (Matrix...)

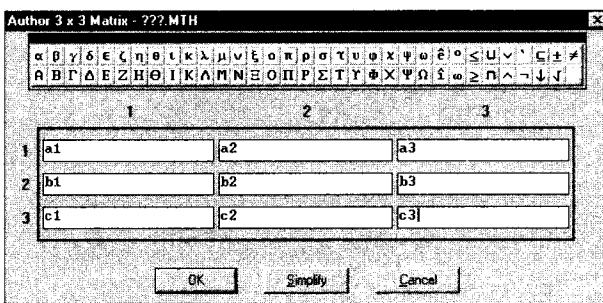
Матрица — это двумерная таблица, состоящая из  $m$  строк и  $n$  столбцов. Эти числа определяют размерность матрицы. Для задания матрицы надо исполнить команду Matrix... и в появившемся окне (рис. 4.41) ввести числа  $m$  и  $n$  (от 1 до 100). Их можно ввести как непосредственно, так и активизируя счетчики числа строк и столбцов.

**Рис. 4.41**  
Окно ввода  
размерности  
матрицы



После ввода размерности матрицы появляется окно ввода элементов матрицы (см. рис. 4.42). Заметим, что матрица с размером  $1 \times 1$  является единственным элементом, а матрица с одной строкой ( $m = 1$ ) — вектором. Вектор может располагаться и вертикально, если  $m > 1$ , а  $n = 1$ .

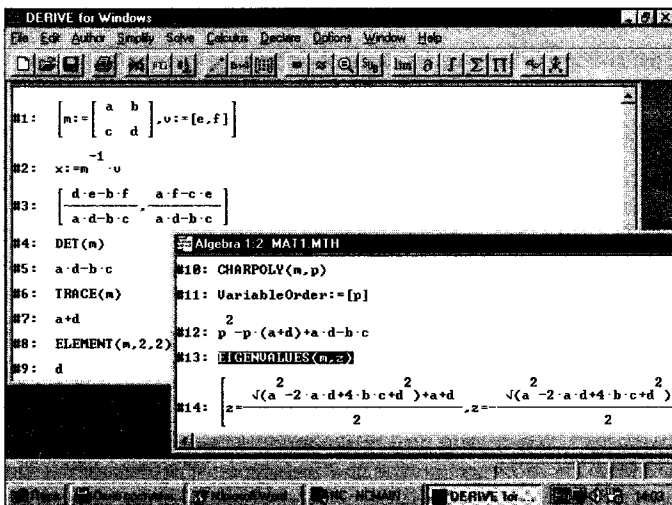
**Рис. 4.42**  
Окно ввода эле-  
ментов матрицы



Если указан ввод больших матриц, справа и внизу окна ввода появляются линейки прокрутки по вертикали и по горизонтали. С их помощью можно ввести невидимые в рабочей части окна элементы. Элементы матриц могут быть числами, именами переменных и символьными выражениями (формулами).

На рис. 4.43 показаны примеры работы с матрицами в системе Derive под Windows. Здесь также подобраны примеры, иллюстрирующие символьные вычисления с матрицами, и примеры вычисления характеристического полинома и собственных значений матриц.

**Рис. 4.43**  
Примеры работы  
с матрицами в  
системе Derive  
под Windows



С дополнительными примерами работы с матрицами вы познакомились в главах 2 и 3. Полезно также обратить внимание на то, что матрицы в Derive являются списками и возможно создание списков в списках.

## 4.7. Преобразования выражений (Simplify...)

Следующим этапом работы с системой является вычисление и преобразование введенных выражений и данных. Позиция главного меню Simplify (Упрощение) выводит подменю с рядом типовых команд символьной математики:

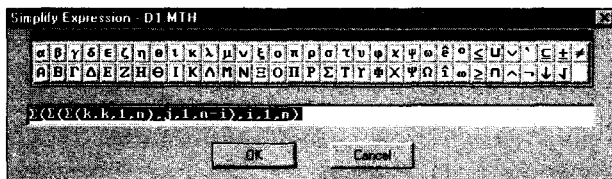
- Basic... (Ctrl + B)** — общее упрощение выражений различного типа;
- Expand... (Ctrl + E)** — расширение полиномиальных выражений;
- Factor... (Ctrl + F)** — факторизация полиномиальных выражений;
- Approximate... (Ctrl + G)** — вычисление выражений в численном виде;
- Substitute for** — подстановки.

Применение для этой позиции имени Simplify не совсем удачно, поскольку под таким именем в системах символьной математики подразумевается вполне конкретная и очень мощная команда упрощения математических выражений. Но одна! В нашей версии Derive так названа общая совокупность команд по преобразованию и вычислению выражений, некоторые из которых отнюдь не упрощают выражения.

### 4.7.1. Общее упрощение выражений (Basic...)

Команда Basic... в нашей версии Derive — это фактически и есть команда Simplify. Обычно эта команда применяется к выделенному выражению или части выражения. Она вызывает появление диалогового окна, показанного на рис. 4.44.

**Рис. 4.44**  
Диалоговое окно  
команды Simplify



Это окно содержит строку с выделенным выражением и наборную панель специальных символов. С их помощью можно корректировать выражение, для которого используется эта команда. Кнопка ОК обеспечивает выполнение команды, тогда как кнопка Cancel позволяет отказаться от этой операции.

Для более четкого представления о функциях команды Simplify на рис. 4.45 показан ряд типовых применений для алгебраических выражений (в основном полиномиальных). Для каждого примера приводится строка с исходным выражением и строка, полученная при применении команды Basic...

Как видно из рис. 4.45, наиболее характерные действия команды Simplify следующие:

- комбинирование цифровых подвыражений — строки 1, 2;
- приведение подобных множителей произведений — строки 3, 4;
- приведение подобных членов сумм — строки 5, 6;
- учет тождеств, содержащих 0, — строки 7, 8;



- учет тождеств, содержащих 1, — строки 9, 10;
- распределение целочисленных показателей степени в произведения — строки 11, 12;
- сокращение на наибольший полиномиальный делитель — строки 13, 14;
- разложение произведений и целочисленных степеней полиномов до выражений, в которых исключена переменная и понижена степень переменной, — строки 15, 16;
- приведение общих знаменателей к таким выражениям, которые позволили бы исключить переменную или понизить степень переменной, — строки 17, 18;
- упрощение выражений без раскрытия высшей степени — строки 19, 20.

Особый интерес представляет упрощение отдельных частей выражения, которое приводит подчас к совсем иным результатам, чем упрощение выражения в целом. Так, на рис. 4.45 в строках 21 и 22 дано упрощение выражения

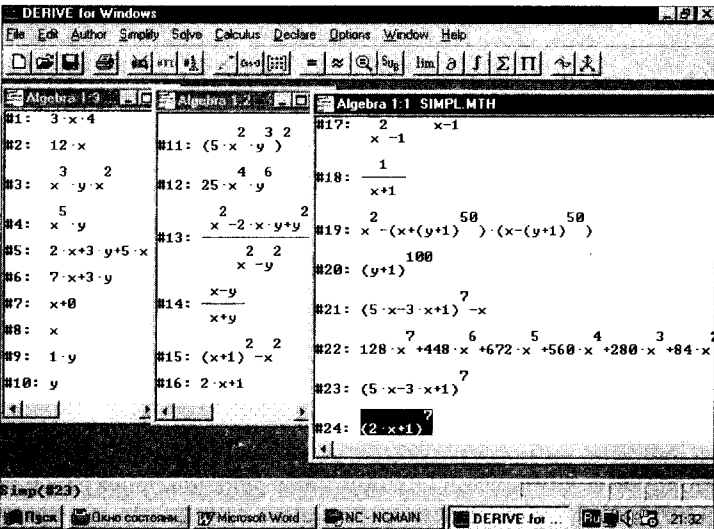
$$(5x - 3x + 1)^7 - x$$

которое создает довольно длинный обычный полином. Но если выделить выражение в скобках седьмой степени и упростить его, то окажется, что исходное выражение упростится к виду

$$(2x + 7)^7 - x$$

Рис. 4.45

Типовые примеры применения команды Simplify

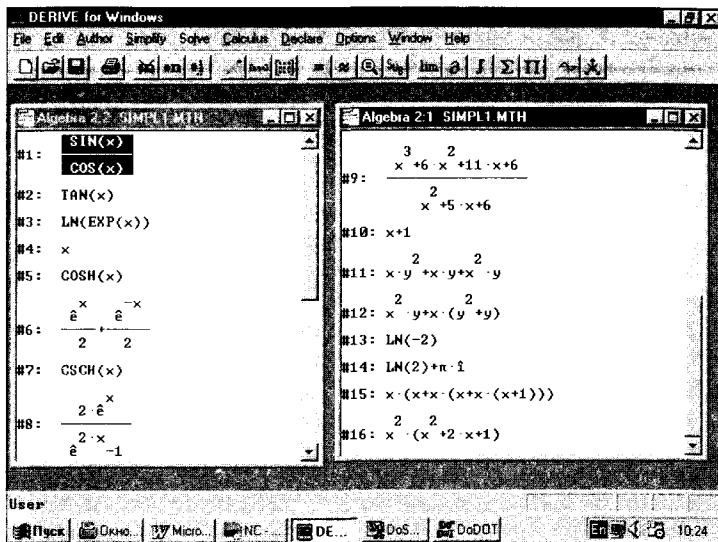


Это еще раз говорит о том, что упрощение выражений — процесс творческий и его результат зависит от подхода к упрощению.

Команда Simplify может применяться для упрощения тригонометрических и алгебраических выражений, содержащих переменные в степени и логарифмы. Ряд примеров применения этой команды для таких выражений представлен на рис. 4.46.

В главах 2 и 3 вы встречали и другие примеры применения команды Simplify, в том числе для упрощения алгебраических и тригонометрических выражений. Результаты таких преобразований зависят от самого направления преобразований, пути управления которым будут обсуждены ниже.

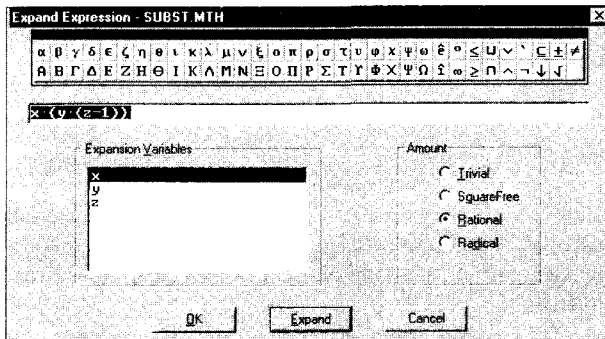
**Рис. 4.46**  
Дополнительные  
примеры приме-  
нения команды  
Simplify



## 4.7.2. Расширение выражений (Expand...)

Расширение выражений чаще всего используется для раскрытия полиномов и максимизации членов алгебраических выражений. Для рациональных выражений оно обычно дает разложение на элементарные дроби. Раскрытием и приведением показательных, логарифмических и тригонометрических выражений команда Expand... не управляет. На рис. 4.47 показано диалоговое окно, которое выводится при использовании команды Expand...

**Рис. 4.47**  
Диалоговое окно  
команды  
Expand...



Как видно из рис. 4.47, в диалоговом окне имеется строка для коррекции расширяемого выражения. Обычно в начале этой строки содержится выделенное выражение, но его можно скорректировать, например, применяя символы наборной панели. При нажатии кнопки ОК вводится строка с функцией EXPAND, а при нажатии кнопки Expand... — строка с уже расширенным выражением. Если расширяемое выражение уже представлено в расширенной форме, команда Expand просто повторяет его.

В окне команды Expand... содержится список переменных, которые входят в расширяемое выражение. С помощью этого списка можно задать раз-

личные виды расширения относительно той или иной переменной. Заметим, что, щелкая мышью по переменным, можно выделить одну или несколько переменных. Такое выделение отменяет ранее установленный приоритет в разложении по переменным (об этом речь пойдет ниже).

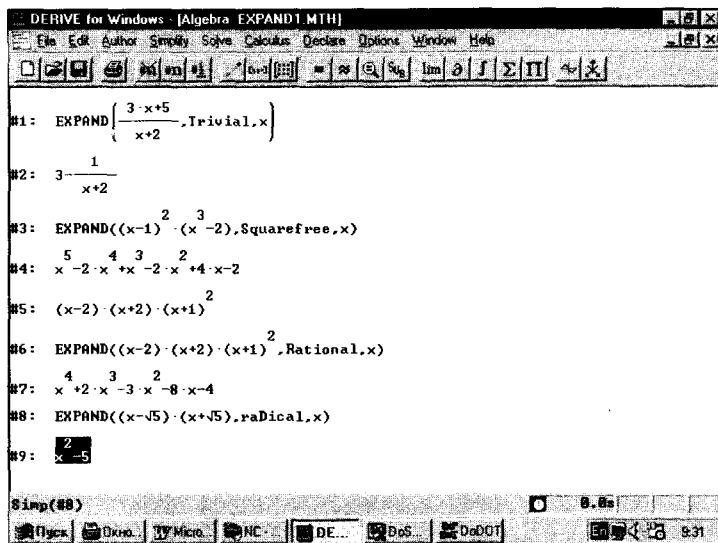
Кроме того, существуют дополнительные возможности для задания опций расширяемого выражения:

- Trivial** — тривиальное;
- SquareFree** — свободное от радикалов;
- Rational** — рациональное;
- Radical** — радикальное.

Для выбора типа выражения в соответствующий кружок устанавливается жирная точка. Проще всего это сделать щелчком мыши или нажатием клавиши Alt и клавиши с подчеркнутой буквой.

На рис. 4.48 показано применение команды Expand...

**Рис. 4.48**  
Примеры применения команды Expand...



Результаты преобразований с помощью команды Expand... зависят от пути преобразований и выбора опции для расширяемого выражения. А теперь просмотрите действие команды Factor для лучшего представления смысла указанных опций.

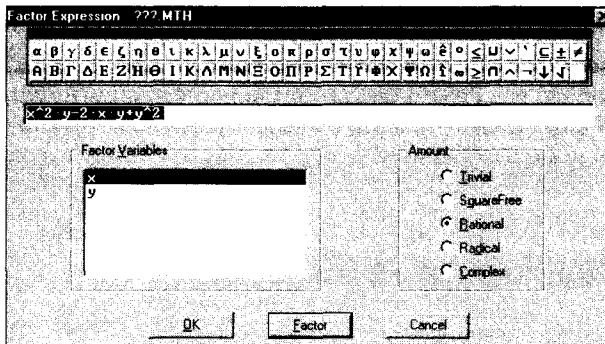
### 4.7.3. Факторизация выражений (Factor...)

Факторизацией в системах символьной математики называют разложение на множители выражений или подвыражений относительно одной или ряда переменных. На рис. 4.49 показано диалоговое окно команды Factor..., которое появляется при ее применении к выделенному выражению.

Это окно подобно окну команды Expand... В нем имеются следующие опции представления выходного выражения:

- Trivial** — тривиальное;
- SquareFree** — свободное от радикалов;

Рис. 4.49

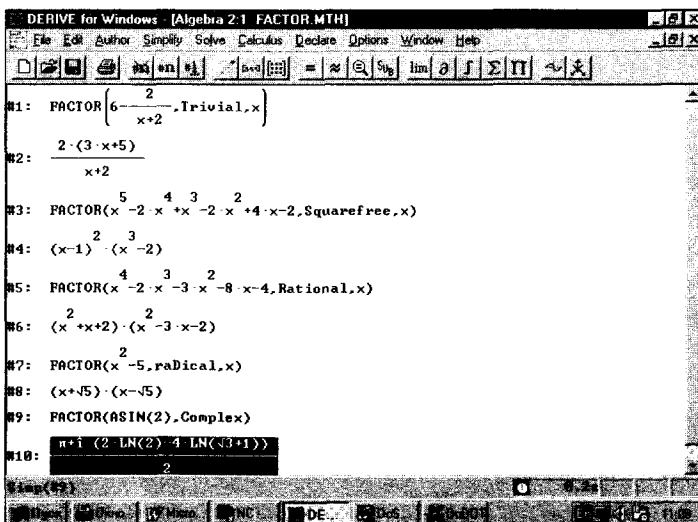
Диалоговое окно  
команды Factor...

- Rational** — рациональное;  
**Radical** — радикальное;  
**Complex** — комплексное.

При тривиальном разложении выражения приводятся к общему знаменателю, числа и переменные с низшими степенями выводятся за скобки. При свободном от радикалов разложении также выполняется тривиальное разложение с выделением множителей в виде степеней сумм или произведений различных степеней сумм. Рациональное разложение осуществляет разложение, свободное от радикалов, с выделением произведений сумм и без введения новых дробных степеней комплексных чисел. При радикальном разложении осуществляется рациональное разложение, но с возможностью представления дробных степеней чисел. При комплексном разложении могут использоваться комплексные числа.

На рис. 4.50 показана факторизация выражений, представленных полиномами и их отношением.

Рис. 4.50

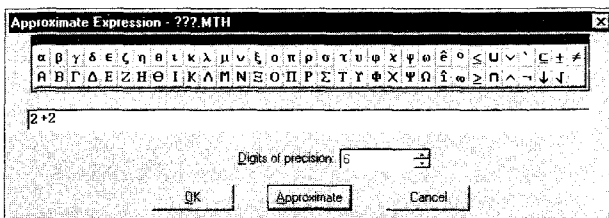
Примеры  
факторизации  
для выражений  
с одной  
переменной

Факторизация совместно с командой расширения Expand дает эффективные методы преобразования полиномиальных выражений. Рекомендуется внимательно просмотреть примеры применения команд Expand... и Factor... и убедиться в своеобразном единстве их противоположностей.

#### 4.7.4. Вычисления выражений в численном виде (Approximate...)

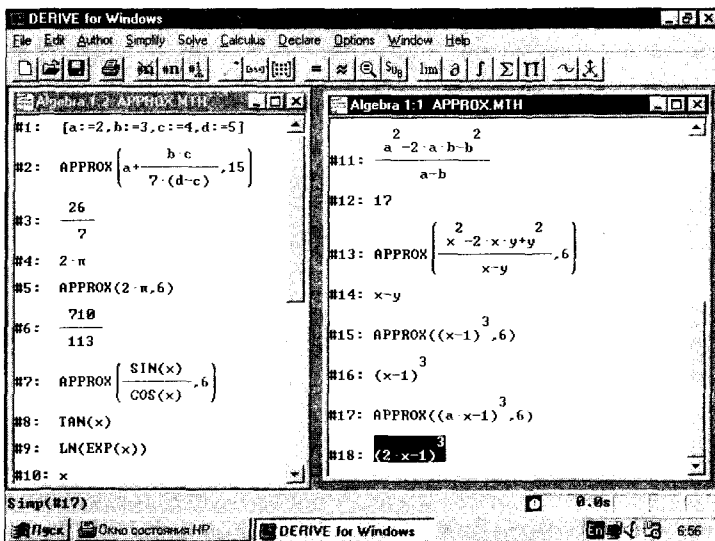
Для получения результатов в численном виде широко используется команда Approximate... Она выводит окно, представленное на рис. 4.51. Как обычно, это окно позволяет редактировать выражение и формировать функцию APPROX при использовании кнопки ОК или результат вычисления отредактированного выражения при использовании кнопки Approximate. Для отказа от операции используется кнопка Cancel.

**Рис. 4.51**  
Окно команды  
Approximate...



Если в выделенном выражении переменные имеют численные значения, команда Approximate... использует их, пытаясь вычислить заданное выражением и представить результат вычисления в виде числа. Рис. 4.52 демонстрирует применение этой команды.

**Рис. 4.52**  
Применение  
команды Approximate...



Если в выражении используются переменные, не имеющие численных значений, команда Approx выполняет функции упрощения выражений.

#### 4.7.5. Подстановки (Substitute for)

Подстановки – еще одна важная возможность систем символьной математики. В Derive под Windows они реализованы двумя командами:

**Substitute for Variable...**

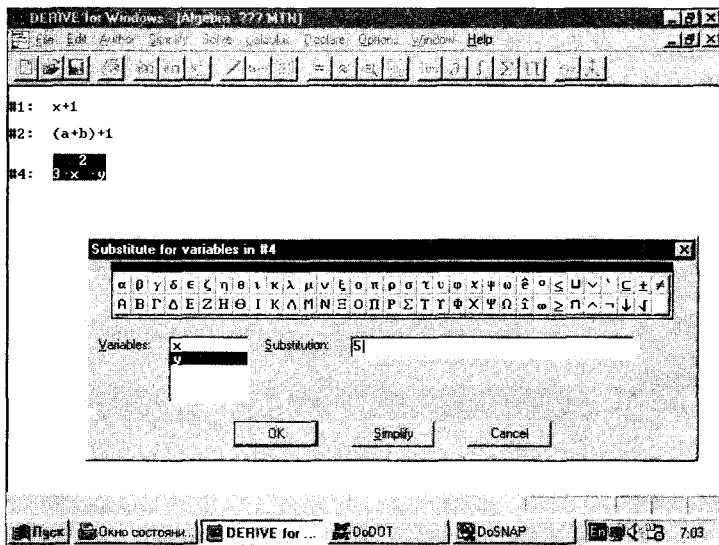
— подстановка для переменной;

**Substitute for Subexpression...**

— подстановка для выделенного подвыражения.

Если выделено все выражение, любой его переменной можно присвоить символьное или численное значение. Это открывает большие возможности в конструировании новых выражений или в сообщении выражениям вида, позволяющего реализовать те или иные математические действия.

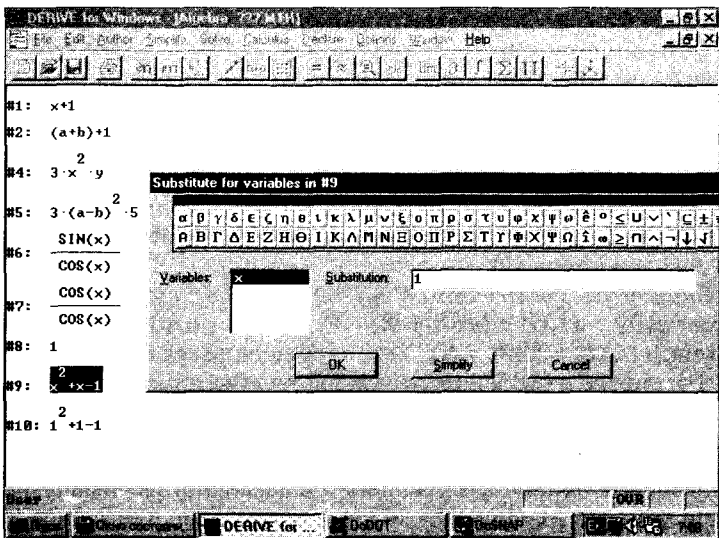
**Рис. 4.53**  
Окно подстановки  
переменной



Команда Substitute for Variable... открывает окно, представленное на рис. 4.53. В этом окне отображается список переменных для выбранного выражения. Выбрав переменную, можно на ее место подставить значение, которое задается в строке набора значений переменной.

На рис. 4.54 показан пример подстановки на место переменной ее численного значения. Заметьте, что тут подстановка производится без упроще-

**Рис. 4.54**  
Пример подста-  
новки на место  
переменной ее  
численного  
значения

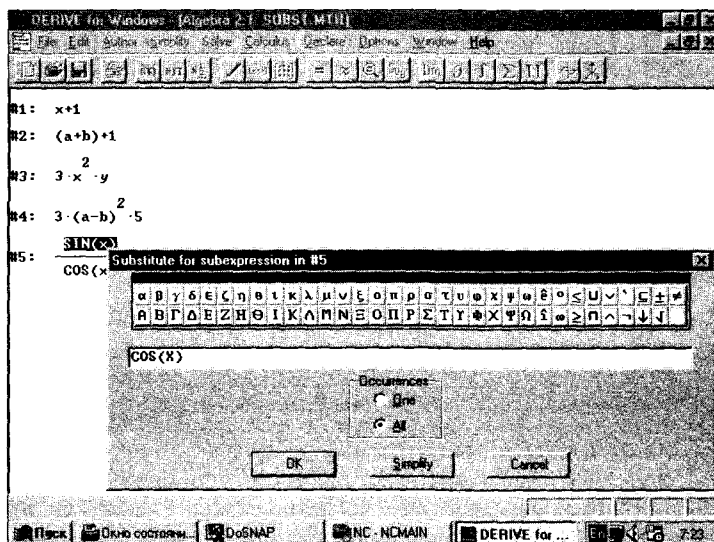


ния выражения, т. е. формально. Если полученное выражение надо вычислить, используйте команды Simplify Basic или Simplify Approximate.

Подстановки, особенно в символьной форме, — излюбленный прием математиков. Они часто сводят трудные и даже невозможные преобразования к вполне возможным.

Подстановка на место одного подвыражения другого выполняется командой Substitute for Subexpression. Эта команда вызывает появление окна, показанного на рис. 4.55. В окне в строке ввода выражения нужно записать новое выражение; в нашем случае функция SIN(x) заменяется на функцию COS(x).

**Рис. 4.55**  
Пример замены одного подвыражения другим



В окне этой команды содержатся две опции Occurrences:

- One** — замена только в одном выделенном подвыражении;
- All** — замена во всех подвыражениях.

Эти опции позволяют выполнить замену или только одного выделенного выражения, или во всех подвыражениях, совпадающих с выделенным.

## 4.8. Решение уравнений и неравенств (Solve)

Для решения уравнений, неравенств и систем с ними Derive под Windows содержит отдельную позицию главного меню — Solve. В ней сосредоточены следующие три команды:

- Algebraically... (Ctrl + Shift + A)** — решение в символьном виде;
- Numerically... (Ctrl + Shift + N)** — решение в численном виде;
- System... (Ctrl + Shift + S)** — решение систем уравнений.

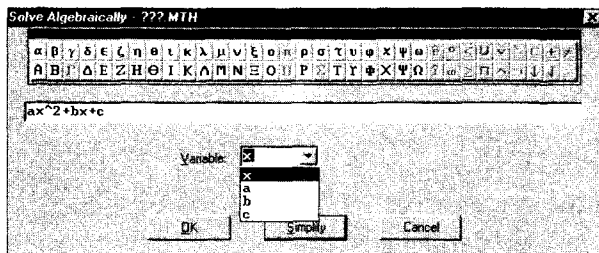
Рассмотрим особенности каждой из этих команд.

### 4.8.1. Решение в символьном виде (Algebraically...)

Для решения одиночных уравнений и неравенств в символьном виде используется команда Algebraically... Она выводит окно для записи уравнения и вы-

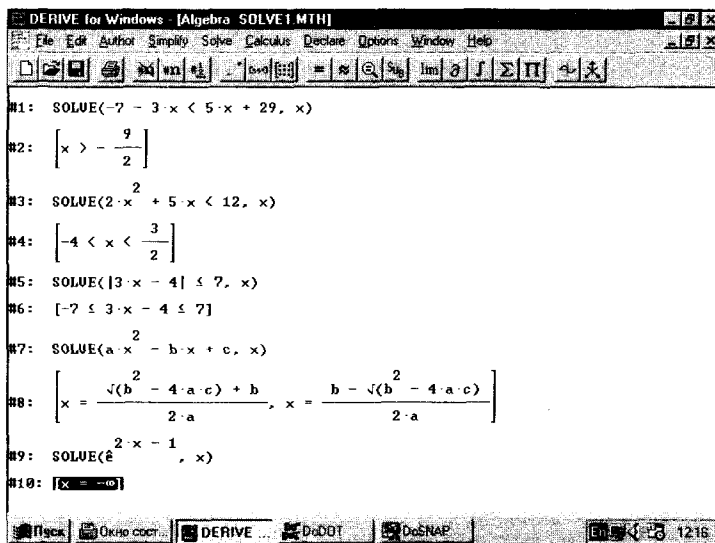
бора из списка переменной, относительно которой решается выбранное или набранное уравнения (см. рис. 4.56).

**Рис. 4.56**  
Окно решения уравнений и неравенств в символьном виде



На рис. 4.57 представлено решение нескольких неравенств и квадратного уравнения с помощью команды Algebraically... В заключительном примере показано, что результатом решения может быть и отрицательная бесконечность.

**Рис. 4.57**  
Примеры решения неравенств и уравнений



Еще несколько примеров применения команды Algebraically... показаны на рис. 4.58. Здесь особенно полезно отметить два последних примера поиска корней полинома — как комплексных для полинома пятой степени, так и действительных для полинома четвертой степени (речь идет не о полиномах вообще, а только о приведенных в примерах).

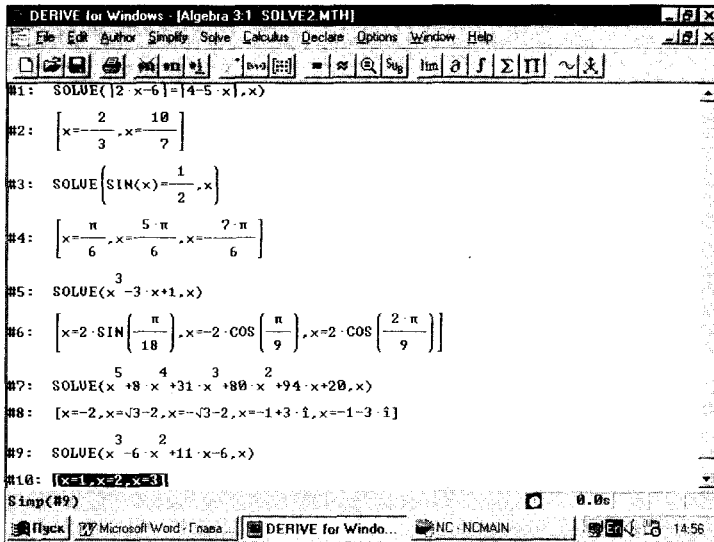
Несколько других примеров применения этой команды были приведены в главах 2 и 3.

#### 4.8.2. Решение в численном виде (Numerically...)

Если необходимо решение уравнения в численном виде, используется команда Numerically... Отличием этого вида решения от уже рассмотренного является не только представление результата в численном виде, но и задание в функции SOLVE пределов, в которых ищется корень (или корни) уравнения.



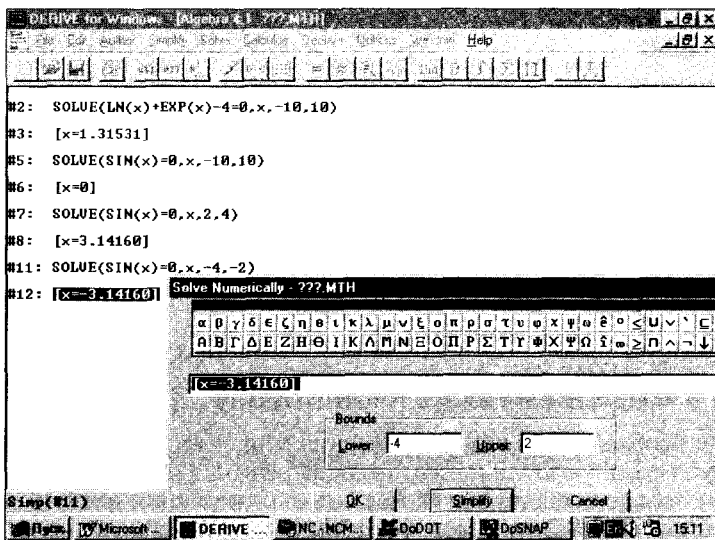
**Рис. 4.58**  
Дополнительные  
примеры реше-  
ния неравенств  
и уравнений



На рис. 4.59 показаны примеры применения команды Numerically и окно этой команды.

Нетрудно заметить, что эта команда создает такой вариант функции SOLVE, когда, помимо самого уравнения и переменной, по которой оно решается, задаются два предела для области, где разыскиваются корни. Уточнением этих пределов можно выделить корень, интересующий вас. Например, для равенства  $\sin(x) = 0$  корни соответствуют  $x = 0$  и  $\pm i\pi$ , где  $i = 1, 2, 3$  и т. д. Тут вначале обнаружен корень для  $x = 0$ . Для поиска других корней (как положительных, так и отрицательных) надо заключать их примерные значения в пределы.

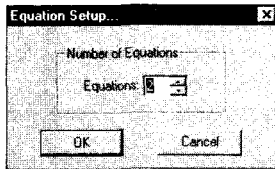
**Рис. 4.59**  
Примеры реше-  
ния уравнений в  
численном виде



### 4.8.3. Решение систем уравнений (System...)

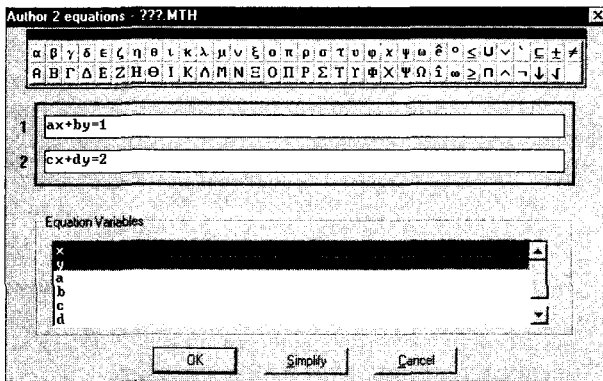
Если необходимо решать системы линейных относительно неизвестных уравнений, следует использовать команду System... При запуске этой команды вначале выводится окно с запросом о числе уравнений в системе (см. рис. 4.60).

**Рис. 4.60**  
Окно ввода числа уравнений



После ввода числа уравнений и подтверждения ввода нажатием кнопки ОК появляется окно для ввода уравнений и указания переменных, относительно которых ищется решение. Окно ввода уравнений показано на рис. 4.61. Следует отметить, что для задания переменных надо щелкнуть мышью по панели, которая находится в окне ввода.

**Рис. 4.61**  
Окно ввода уравнений

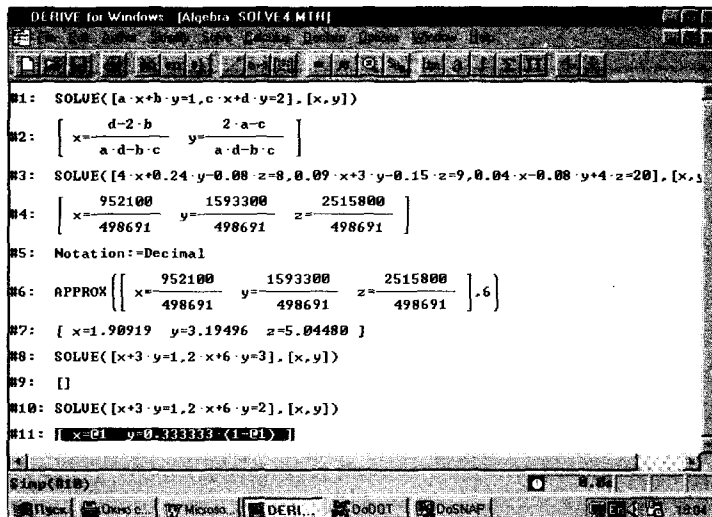


После ввода уравнений и указания переменных формируется функция SOLVE для решения введенной системы. Как система уравнений, так и переменные, относительно которых ищется решение, представляются в виде вектора. Для решения системы достаточно исполнить команду Simplify Basic... Возможно решение и без создания функции SOLVE. Для этого, не выходя из окна ввода уравнений, достаточно нажать кнопку Simplify. Для отказа от решения системы уравнений надо нажать клавишу Cancel.

На рис. 4.62 приведен ряд примеров решения систем линейных уравнений. Относительно других переменных уравнения могут быть нелинейными. Обратите внимание на два последних примера. В предпоследнем примере задана несовместная система уравнений. Поскольку тут решения не существует, выдается пустой вектор в виде квадратных скобок.

Система уравнений в последнем примере — вырожденная, поскольку уравнения не являются независимыми друг от друга. Если вырожденная система совместна, в решении появляются одно или несколько произвольных значений вида @n, где n — целое число. Число таких значений во множестве решений равно общему числу уравнений за вычетом числа независимых уравнений.

**Рис. 4.62**  
Примеры  
решения систем  
уравнений



## 4.9. Основные виды вычислений (Calculus)

В позиции Calculus главного меню сосредоточены наиболее важные команды для выполнения вычислений, относящихся к фундаментальным приложениям математического анализа. Подменю этой позиции является перечнем таких вычислений:

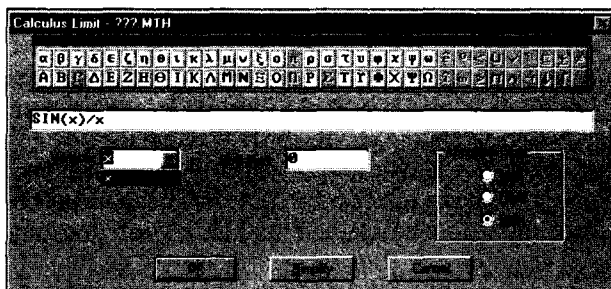
- Limit...** — вычисление пределов функций;
- Differentiate...** — вычисление производных функций;
- Taylor series...** — разложение функций в ряд Тейлора;
- Integrate....** — вычисление неопределенных и определенных интегралов;
- Sum...** — вычисление сумм рядов;
- Product...** — вычисление произведений рядов;
- Vector...** — представление ряда в виде вектора.

Рассмотрим работу с этими командами более подробно.

### 4.9.1. Вычисление пределов функций (Limit...)

Для вычисления предела функции можно ввести ее как отдельное выражение и, выделив, исполнить команду Limit. Эту команду можно исполнить и не вводя функцию заранее; ее придется ввести в наборную панель окна команды Limit..., показанного на рис. 4.63.

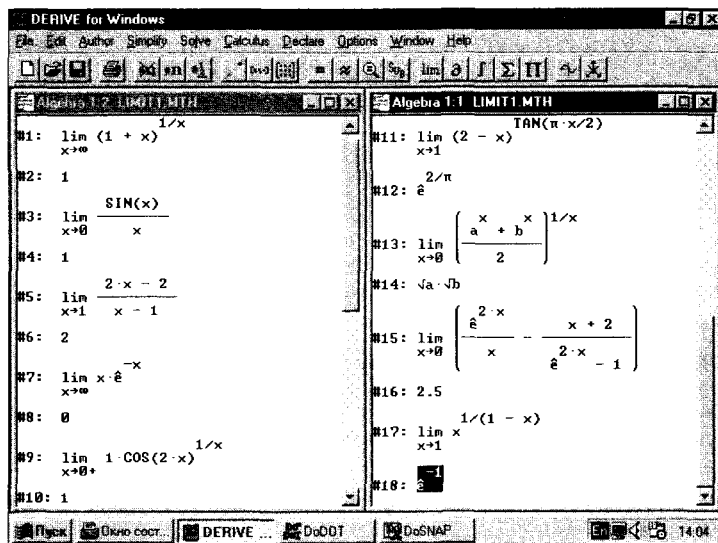
**Рис. 4.63**  
Окно команды  
Limit...



Это окно позволяет также выбрать переменную, относительно которой ищется предел, и установить ее значение (Limit Point). Кроме того, можно установить тип изменения переменной при поиске предела функции (Approach From): подход слева (Left), справа (Right) и с двух сторон (Both). Это необходимо, в частности, когда ищутся пределы разрывных функций.

Как обычно, окно команды Limit... по завершении установок позволяет сразу вычислить предел, нажав кнопку Simplify либо введя функцию LIMIT, для вычисления которой необходимо исполнение команды Simplify Basic... (или Approximate... в случае нахождения предела в численной форме). На рис. 4.64 показан ряд примеров вычисления пределов различных функций.

**Рис. 4.64**  
Вычисление пределов различных функций



## 4.9.2. Вычисление производных функций (Differentiate...)

Для вычисления производных функций используется команда Differentiate... Она выводит окно (рис. 4.65), в котором задаются дифференцируемая функция, переменная, по которой осуществляется дифференцирование, и порядок искомой производной.

Затем кнопка ОК создает функцию дифференцирования DIFF. Можно также найти производную без этой функции, активизировав клавишу Simplify. На рис. 4.66 представлены примеры дифференцирования различных функций.

**Рис. 4.65**  
Окно команды дифференцирования

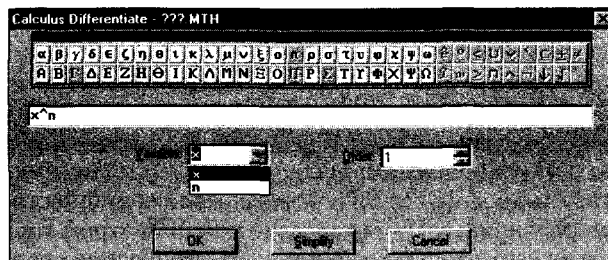
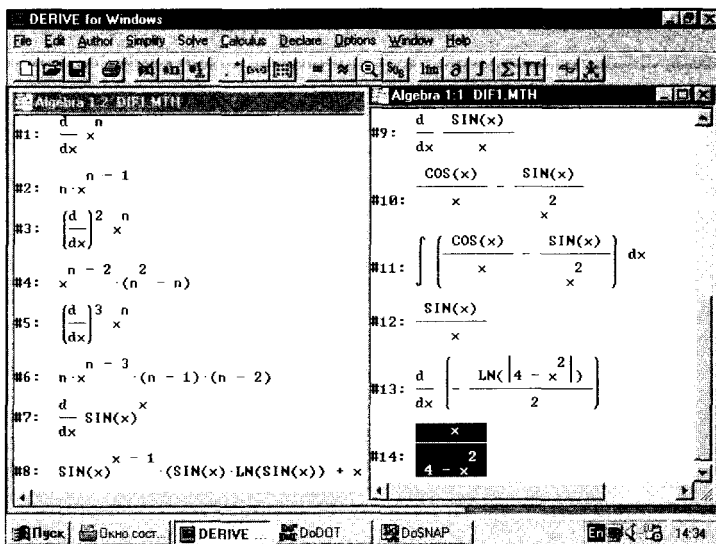


Рис. 4.66

Примеры дифференцирования различных функций



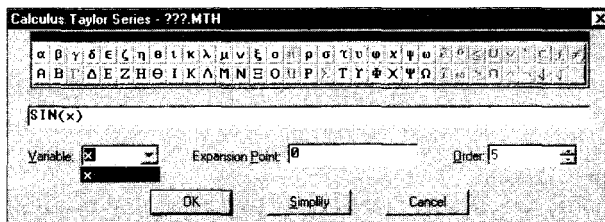
Обратите внимание на то, что нередко операцию дифференцирования легко проверить интегрированием полученного выражения.

### 4.9.3. Разложение функций в ряд Тейлора (Taylor series...)

Разложение функций в ряд Тейлора — одна из мощных команд, позволяющих наглядно проиллюстрировать замену различных функциональных зависимостей их приближенным представлением. Для выполнения этой операции достаточно исполнить команду Taylor series..., которая выводит окно, представленное на рис. 4.67.

Рис. 4.67

Окно команды разложения функций в ряд Тейлора

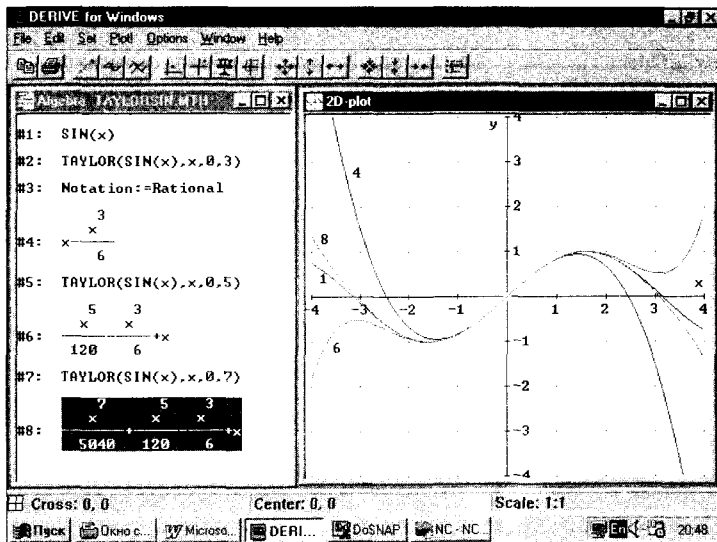


Здесь задаются разлагаемая в ряд функция и переменная (Variable), относительно которой выполняется разложение, указываются значение этой переменной в точке разложения (Expansion Point) и максимальный порядок членов разложения (Order). Нажатие клавиши Simplify сразу возвращает результат разложения, тогда как нажатие клавиши ОК создает функцию TAYLOR со всеми необходимыми для ее вычисления параметрами.

На рис. 4.68 показан наглядный пример разложения функции SIN(x) в ряд Тейлора с максимальным порядком членов разложения 3, 5 и 7.

Здесь для исходной функции и всех ее разложений построены графики функции; номера у кривых соответствуют номерам строк, в которых записаны соответствующие выражения. Графики дают прекрасное представление о том,

**Рис. 4.68**  
Пример разложения синусоиды в ряд Тейлора

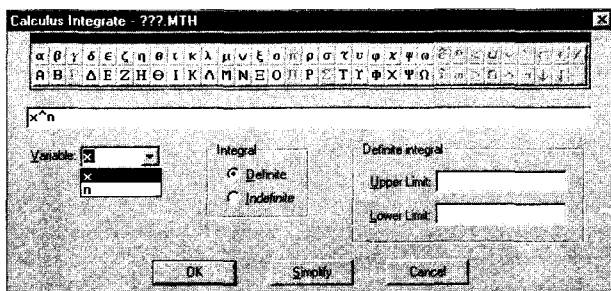


как по мере роста порядка разложения растет его точность. Этот пример — хорошая иллюстрация к технике визуализации вычислений, проводимых с помощью Derive.

#### 4.9.4. Вычисление неопределенных и определенных интегралов (Integrate...)

Для вычисления неопределенных (без указания пределов интегрирования) и определенных интегралов используется команда Integrate... . На рис. 4.69 показано окно этой функции, позволяющее задать подынтегральную функцию, верхний (Upper Limit) и нижний (Low Limit) пределы интегрирования для определенных интегралов и переменную, по которой осуществляется интегрирование. Кроме того, надо указать, какой интеграл вычисляется — определенный (Definite) или неопределенный (InDefinite).

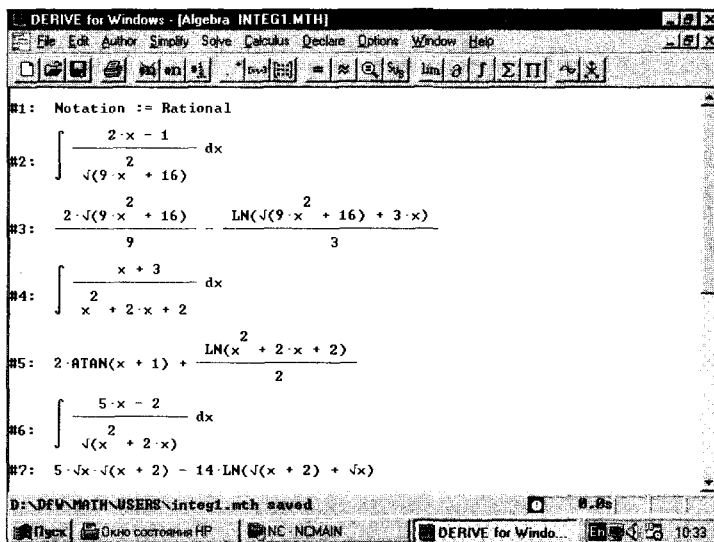
**Рис. 4.69**  
Окно команды Integrate...



Нажатие кнопки OK создает функцию INTEGRATE со всеми нужными для ее вычисления параметрами. Клавиша Simplify позволяет вычислить интеграл сразу и возвращает его значение (символьное или численное).

Поскольку вычисление интегралов — одна из самых распространенных областей применения математических систем, рассмотрим применение Derive под Windows на достаточно большом количестве примеров. На рис. 4.70 показано вычисление однократных простых интегралов с полиномиальными и рациональными подинтегральными функциями.

**Рис. 4.70**  
Примеры вычисления простых интегралов с рациональными подинтегральными функциями



На рис. 4.71 показано несколько примеров вычисления интегралов, у которых подинтегральные функции — алгебраические выражения, содержащие логарифмы и экспоненциальные функции.

Дополнительные примеры вычисления неопределенных интегралов даны на рис. 4.72.

**Рис. 4.71**  
Примеры вычисления неопределенных интегралов с алгебраическими выражениями

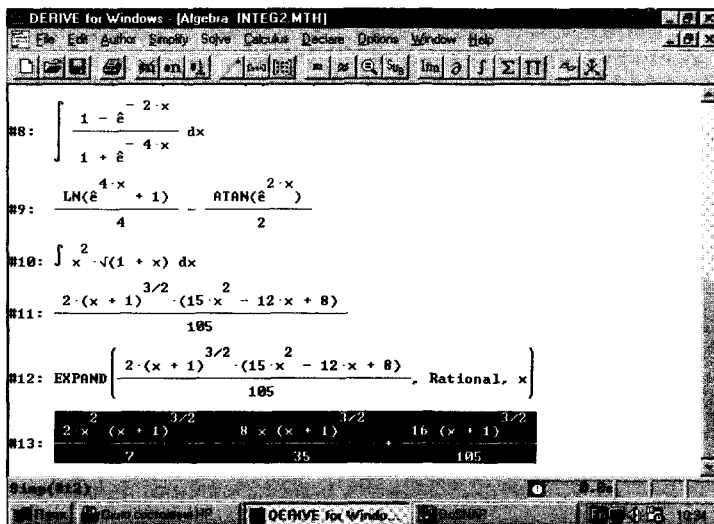


Рис. 4.72

Дополнительные  
примеры  
вычисления  
неопределенных  
интегралов

DERIVE for Windows - [Algebra 3 1 INTEG3 MTH]

#1: Notation := Rational

#2: 
$$\int \frac{18 + 11 \cdot x - x^2}{(x-1) \cdot (x+1) \cdot (x^2 + 3 \cdot x + 3)} dx$$

#3: 
$$-3 \cdot \sqrt{3} \cdot \text{ATAN}\left(\frac{\sqrt{3} \cdot (2 \cdot x + 3)}{3}\right) + \frac{\text{LN}(x^2 + 3 \cdot x + 3)}{2} + 2 \cdot \text{LN}(x-1) - 3 \cdot \text{LN}(x+1)$$

#4: 
$$\int x^2 \cdot \text{COS}(x) dx$$

#5: 
$$2 \cdot x \cdot \text{COS}(x) + (x^2 - 2) \cdot \text{SIN}(x)$$

#6: 
$$\int e^x \cdot \text{SIN}(x) dx$$

#7: 
$$e^x \left( \frac{\text{SIN}(x)}{2} - \frac{\text{COS}(x)}{2} \right)$$

Stamp(#6)

На практике большие трудности вызывает вычисление несобственных интегралов, имеющих либо бесконечные пределы, либо особенности функции в заданных пределах интегрирования. Многочисленные примеры вычисления таких сложных интегралов (порой их внешний вид обманчиво прост) приведены на рис. 4.73.

И наконец, на рис. 4.74 представлены примеры вычисления кратных (двойных и тройных) интегралов, как неопределенных, так и определенных.

Из приведенных примеров можно понять, что Derive очень неплохо справляется с вычислением интегралов любого вида, у которых подынтегральная функция и результат могут быть представлены выражениями с элементарными функциями. Для вычисления интегралов, дающих представле-

Рис. 4.73

Примеры  
вычисления  
несобственных  
интегралов

DERIVE for Windows

Algebra 1 1 INTEG1 MTH

#1: 
$$\int_1^2 \frac{1}{\sqrt{x-1}} dx$$

#2: 2

#3: 
$$\int_0^1 x \cdot \text{LN}(x) dx$$

#4: 
$$-\frac{1}{4}$$

#5: 
$$\int_1^{\infty} \frac{1}{x \cdot \sqrt{x-1}} dx$$

#6: 
$$\frac{\pi}{2}$$

Algebra 1 1 INT1 MTH

#7: 
$$\int_0^{\infty} x \cdot \text{EXP}(-x) dx$$

#8: 1

#9: 
$$\int_0^{\infty} x \cdot e^{-x^2} dx$$

#10: 
$$\frac{1}{2}$$

#11: 
$$\int_0^{\infty} \text{SIN}(x) dx$$

#12: 
$$\text{SIN}(\infty) + 1$$

Algebra 1 2 INTEG4 MTH

#13: 
$$\int_{-\infty}^{\infty} \frac{x}{x^2 + 1} dx$$

#14: ?

#15: 
$$\int_{-\infty}^{\infty} \frac{1}{x^2 + 6 \cdot x + 12} dx$$

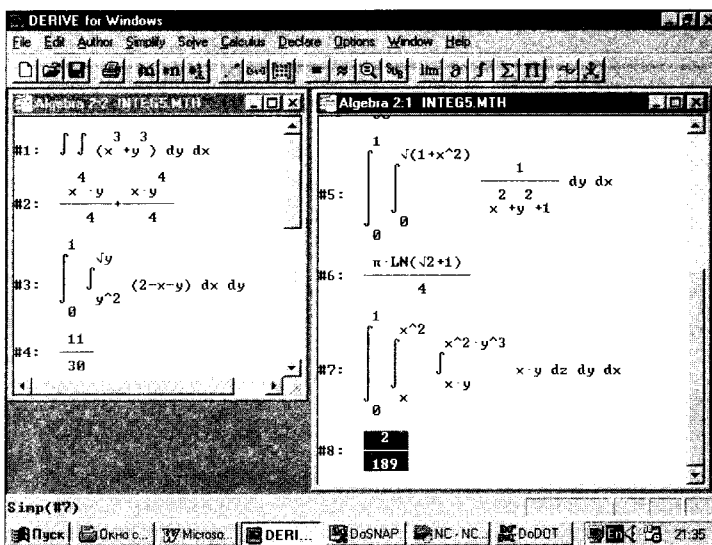
#16: 
$$\frac{\sqrt{3} \cdot \pi}{3}$$

#17: 
$$\int_{-\infty}^{\infty} x \cdot e^{-x^2} dx$$

#18: 0



**Рис. 4.74**  
Вычисление  
кратных  
интегралов

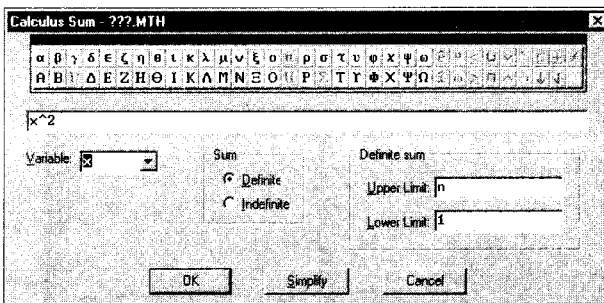


ние в виде специальных математических функций, Derive приспособлена слабо. Это иногда возможно при использовании библиотек расширения Derive.

#### 4.9.5. Вычисление сумм рядов (Sum...)

Для вычисления сумм рядов используется команда Sum... . Ее исполнение выводит окно, показанное на рис. 4.75. В нем задаются функция, значения которой суммируются, индексная переменная и верхний и нижний пределы ее изменения. Кроме того, можно задать определенную и неопределенную сумму; в последнем случае вычисляется ряд с бесконечным числом членов.

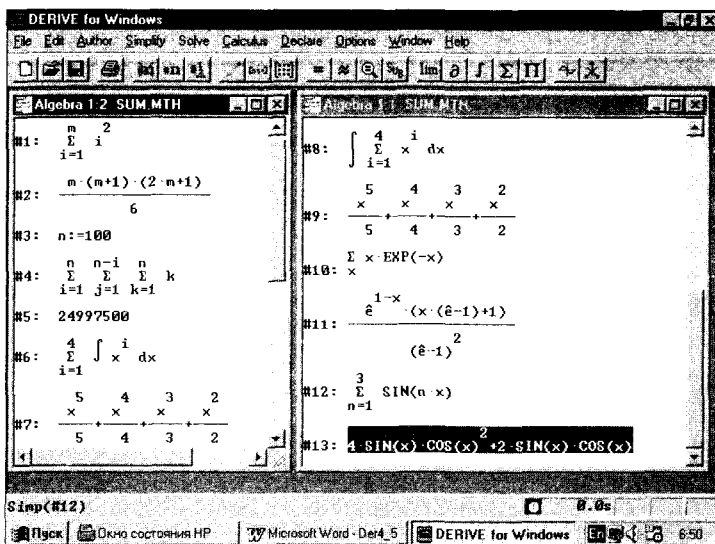
**Рис. 4.75**  
Окно вычисления  
сумм рядов



На рис. 4.76 представлено большое количество примеров вычисления суммы рядов. Нетрудно заметить, что Derive способна вычислять как символичные, так и численные значения сумм. Особо стоит обратить внимание на возможность вычисления сумм рядов, членами которых являются производные и интегралы, а также на вычисление производных и интегралов, содержащих суммы.

Derive успешно справляется с вычислением сумм, на которых спотыкаются более мощные математические системы. К примеру, MathCAD PLUS 6.0

**Рис. 4.76**  
Примеры  
вычисления  
сумм рядов

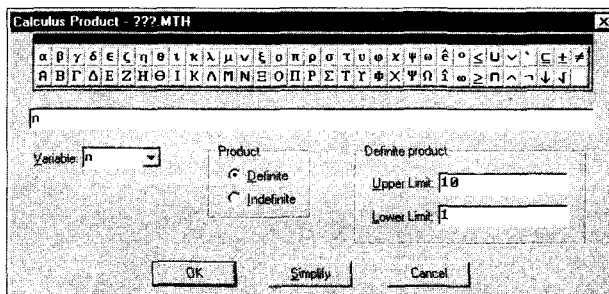


дает ошибку при вычислении тройной суммы в строках 3–5 примера на рис. 4.76, тогда как Derive вычисляет сумму верно.

#### 4.9.6. Вычисление произведений рядов (Product...)

Вычисление произведений членов ряда вычисляется с помощью команды Product... . Окно этой команды показано на рис. 4.77. В нем задаются функция, значения которой перемножаются, индексная переменная и пределы ее изменения.

**Рис. 4.77**  
Окно задания  
функции произве-  
дения рядов

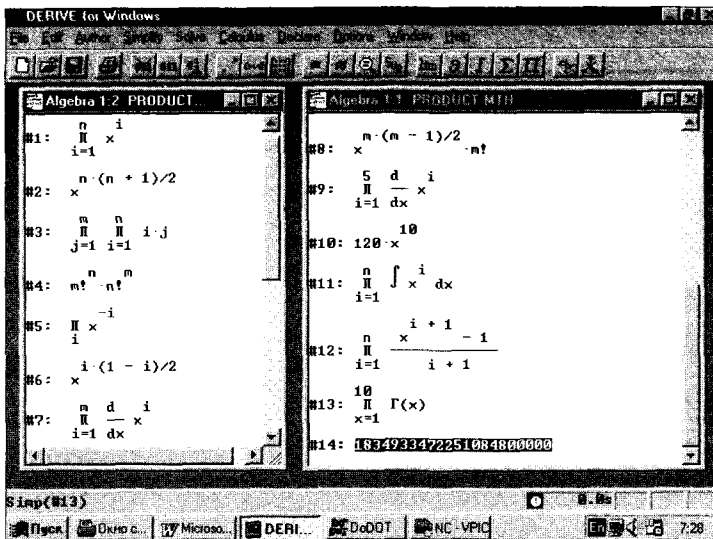


Можно также задать произведение с неопределенным числом членов и определенным; индексная переменная в последнем случае меняется до бесконечного значения.

На рис. 4.78 представлены примеры применения функции произведения рядов. Здесь также полезно обратить внимание на то, что вычисляются произведения как в аналитическом, так и численном виде.

Вычисления сумм и произведений рядов широко используются в специальных математических функциях, имеющих представления через них.

**Рис. 4.78**  
Примеры  
вычисления  
сумм рядов



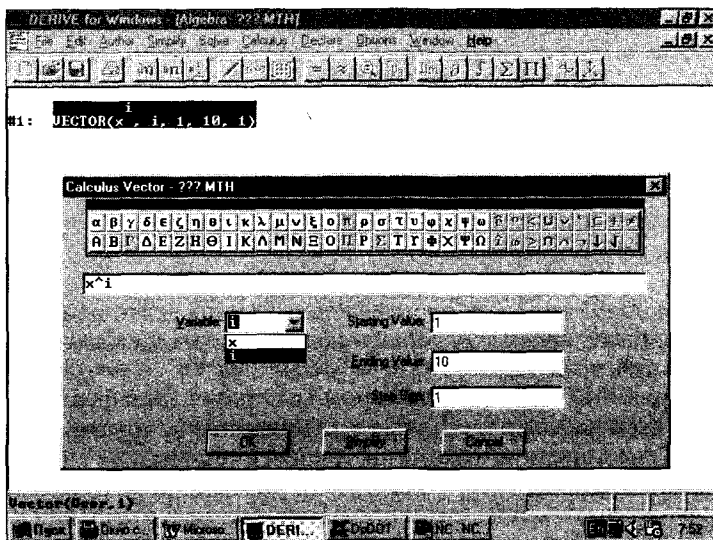
#### 4.9.7. Представление ряда в виде вектора (Vector...)

Функция VECTOR, обеспечивающая генерацию компонентов векторов, уже рассматривалась при обсуждении векторных операций. Как и другие функции, она может генерироваться с помощью специальной команды Vector... . Эта команда вызывает окно, показанное на рис. 4.79.

Помимо задания функции и индексной переменной, в этом окне задаются:

- Starting Value** — начальное значение индексной переменной;
- Ending Value** — конечное значение индексной переменной;
- Step Size** — шаг изменения индексной переменной.

**Рис. 4.79**  
Окно задания  
функции VECTOR



Примеры использования функции VECTOR неоднократно приводились выше. Поэтому ограничимся лишь такой областью применения этой функции, как реализация программных средств, называемых циклами.

## 4.10. Декларация переменных и функций (Declare)

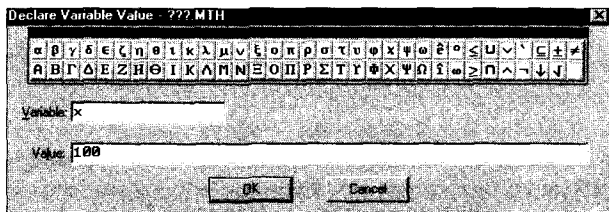
Derive допускает применение необъявленных заранее переменных. Такие переменные наиболее часто применяются в символьных операциях над выражениями. Однако нередко для решения задач (прежде всего численных) необходимо присваивать переменным вполне определенный статус и наделять их конкретными значениями. Кроме того, поскольку набор встроенных в ядро Derive функций ограничен, часто возникает необходимость задавать функции одной или ряда переменных. Все эти возможности (а также задание формата ввода и вывода) предоставляет подменю позиции Declare главного меню. Оно содержит следующие команды:

- Variable Value...** — декларация переменной и присвоение ей значения;
- Variable Domain...** — декларация ранжированной переменной;
- Function Definition...** — задание функции пользователя;
- Algebra State** — вызов подменю управления форматом ввода/вывода.

### 4.10.1. Декларация переменной и присваивание ей значений (Variable Value...)

Для объявления (декларации) переменной и присваивания ей численного или символьного значения используется команда Variable Value. Она выводит окно, показанное на рис. 4.80. С помощью этого окна можно задать имя переменной и ее значение. Заметим, что имя переменной может задаваться как доступными знаками на панели знаков этого окна, так и набором латинских букв.

**Рис. 4.80**  
Окно декларации переменной



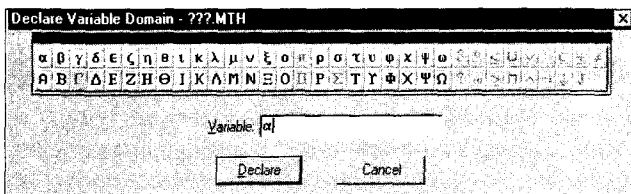
Мы уже неоднократно пользовались заданием переменных. Напомним, что задание переменной сводится к указанию ее имени, знака присваивания := и значения переменной после него. Например, для переменной  $x$  со значением 100, объявленной в окне, показанном на рис. 4.80, будет создана очередная строка с выражением в ней вида  $X := 100$ .

### 4.10.2. Декларация области определения переменной (Variable Domain...)

Derive позволяет использовать обширный класс значений или областей определения переменных. Переменные могут быть с целочисленными, рациональ-

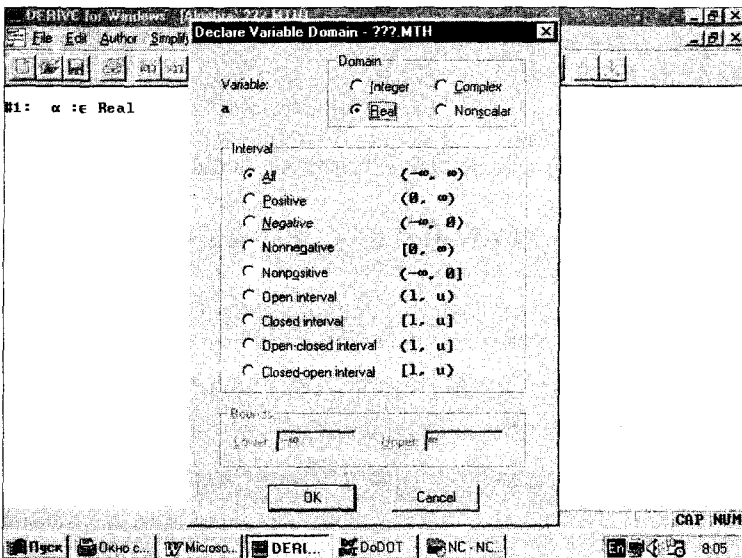
ными, комплексными и иными значениями. Для задания (декларации) области определения некоторой переменной служит команда Variable Domain... Вначале она создает окно декларации самой переменной, показанное на рис. 4.81.

**Рис. 4.81**  
Окно задания переменной с декларированной областью определения



После задания переменной выводится окно с указанием возможных типов переменных и областей их определения (см. рис. 4.82).

**Рис. 4.82**  
Окно декларации области заданной переменной



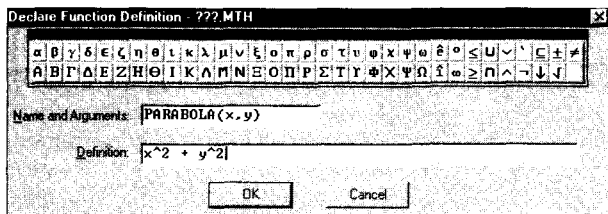
В левом верхнем углу окна выводится имя переменной, для которой декларируются области определения. Их возможные варианты указаны в окне и достаточно очевидны. Всего можно задать четыре типа переменных и девять областей их определения.

### 4.10.3. Задание функции пользователя (Function Definition...)

Функции пользователя в системе Derive под Windows задаются командой Function Definition... (Определение функций). Эта команда выводит окно, показанное на рис. 4.83.

Здесь достаточно задать необходимую функцию и подтвердить это нажатием кнопки ОК. В результате заданная функция будет введена в очередную строку документа и станет доступна для применения.

**Рис. 4.83**  
Окно задания  
функции  
пользователя



#### 4.10.4. Подменю управления форматом ввода/вывода (Algebra State)

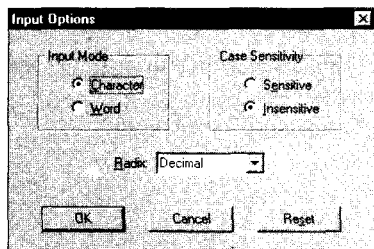
Позиция Algebra State в подменю позиции Declare главного меню выводит свое подменю со следующими командами:

- Input... (Ctrl + I)** — выводит окно опций управления форматом ввода;
- Output... (Ctrl + J)** — выводит окно опций управления форматом вывода;
- Simplification...** — выводит окно опций упрощения;
- Reset All** — отмена всех изменений.

Эти опции действуют глобально до отмены всех изменений командой Reset All.

Команда Input создает окно, показанное на рис. 4.84. Это окно вводит несколько опций ввода, устанавливаемых жирной точкой в соответствующем кружке.

**Рис. 4.84**  
Окно установки  
опций ввода

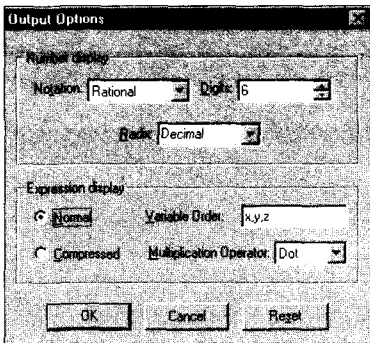


Опций режима ввода Input Mode две: Character — воспринимает ввод слитных символов по отдельности и Word — воспринимает ввод слитных символов как слово. Это имеет значение при вводе идентификаторов (имен) переменных. В первом случае хуz воспринимается как умножение x на y и на z, а во втором случае — как ввод переменной с именем хуz. Другая пара опций группы Case Sensitivity устанавливает чувствительность к регистрам переключения размеров символов или ее отсутствие. По умолчанию установлены ввод слитных символов как их перемножение и отсутствие чувствительности к регистрам.

Кроме того, в этом окне имеется переключатель основания чисел Radix с четырьмя позициями: Binary — двоичное основание, Octal — восьмеричное основание, Decimal — десятичное основание и Hexadecimal — шестнадцатеричное. По умолчанию установлено десятичное основание. Клавиши внизу окна выполняют следующие действия: OK — фиксирует ввод, Cancel — отменяет изменение опций ввода и Reset — уничтожает сделанные изменения и задает значения опций, принятые по умолчанию.

Окно установки опций вывода показано на рис. 4.85. Оно содержит две группы опций: Number display — для отображения чисел и Expression display — для отображения выражений.

**Рис. 4.85**  
Окно установки  
опций вывода

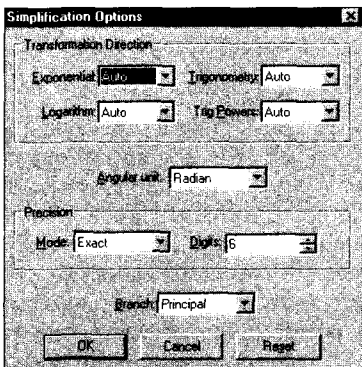


В опции отображения чисел входят установки формата чисел (десятичный — Decimal, смешанный — Mixed, рациональный — Rational и комплексный — Complex), количества цифр числа Digits (по умолчанию 6) и основания (двоичного, восьмеричного, десятичного и шестнадцатеричного).

Опции представления выражений содержат установку вывода в нормальной (Normal) и сжатой (Compesed) форме, установку порядка переменных (по умолчанию x, y, z) и установку знака операции умножения (символом \* Asterisk, точкой Dot и пробелом Implicit).

Окно установки опций упрощения (рис. 4.86) устанавливает направления преобразования (группа опций Transformation Direction), единицы представления углов (Angular unit), точность (Precision), число цифр численных данных (Digits) и выбор ветви преобразований (Branch).

**Рис. 4.86**  
Окно установки  
опций упрощения



Направления преобразования задаются для экспоненциальных (Exponential), тригонометрических (Trigonometry) и логарифмических (Logarithm) функций трех видов: Auto — автоматически, Collect — сбор и Expand — расширение. Для опций группы тригонометрических степеней Trig Power задаются значения Auto Cosines (преобразование к косинусам) и Sines (преобразование к синусам).

Опции группы точности (Precision) задают режимы (Mode): Approximate, Exact и Mixed. Для ветви преобразования задаются опции Principal, Real и Any.

## 4.11. Установка опций (Options)

Некоторые параметры системы (например, цвет символов и фона окна, параметры печати и пр.) неизменны и задаются с помощью команд, размещенных в подменю позиции Options главного меню:

- Color** — установка опций цвета;
- Printing...** — установка опций печати;
- The Entry Dialog** — включение/выключение диалога.

Рассмотрим эти опции более детально.

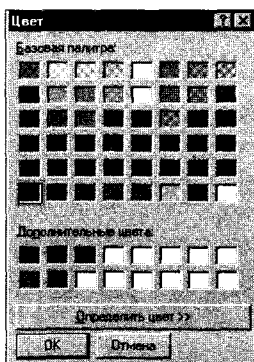
### 4.11.1. Установка опций цвета (Color)

Derive содержит унифицированное окно для установки цвета, которое используется для изменения и добавления цветов во всех опциях, связанных с цветом. Так, команда Color выводит подменю для установки цветов следующих объектов окна выражений:

- Windows Text...** — установка цвета текста;
- Windows Background...** — установка цвета фона;
- Highlight Text...** — установка цвета выделенного текста;
- Highlight Background...** — установка цвета фона выделенного текста.

Все эти команды выводят указанное окно вначале в упрощенном виде, показанном на рис. 4.87. Тут можно задавать цвет выбором его из палитры ряда дискретных цветов. В большинстве случаев этого вполне достаточно.

**Рис. 4.87**  
Окно выбора  
цветов в основ-  
ном варианте

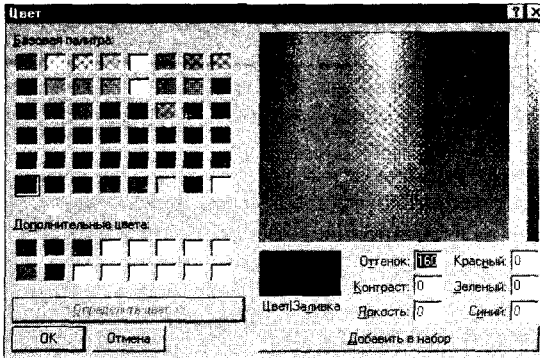


Однако если вы любитель необычных сочетаний цветов, например промежуточных между теми, которые выбираются из палитры дискретных цветов, вы можете задать дополнительные цвета, расширив окно в основном варианте. Для этого надо нажать кнопку “Определить цвет” (опять-таки русскоязычные надписи говорят о том, что это окно принадлежит русской версии Windows 95, а не системе Derive). Окно расширится и примет вид, показанный на рис. 4.88.

Теперь в расширенном окне появляется панель с цветами, плавно переходящими от одного к другому. С помощью жирного креста можно взять пробу любого цвета и задать дополнительные цвета. Их можно задавать также, смешивая три основных цвета (красный Red, зеленый Green и синий Blue) в любой пропорции с помощью ползунков регулировки насыщенности этих цветов.



**Рис. 4.88**  
Окно выбора  
цветов в расши-  
ренном варианте

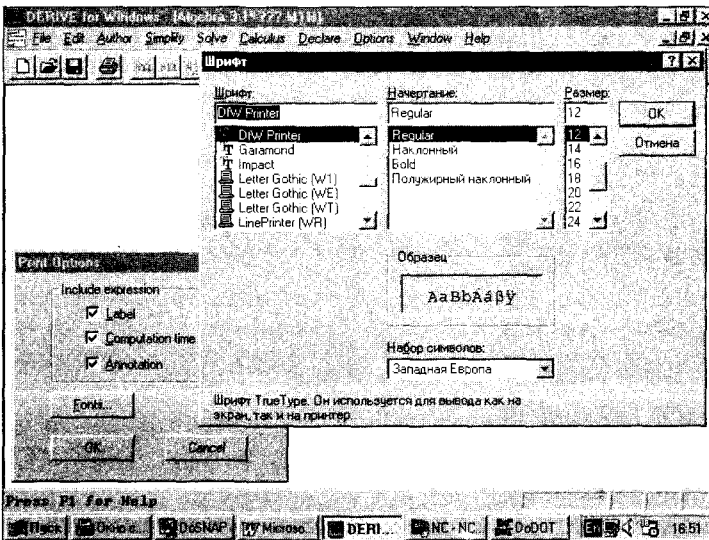


Надо отметить, что это окно используется и в многочисленных других опциях и установках цвета, например для графических окон. Там мы его описывать уже не будем.

#### 4.11.2. Установка опций печати (Printing...)

Команда Options Printing служит для установки опций печати окна выражений. Она выводит окно, показанное на рис. 4.89.

**Рис. 4.89**  
Окно установки  
опций печати ок-  
на выражений



Здесь можно установить три вида дополнительных объектов для печати:

- Label** — печать меток, т. е. номеров строк;
- Communication time** — печать времени вычислений;
- Annotation** — печать аннотаций к выражениям.

Для установки этих опций достаточно поставить с помощью мыши птичку в соответствующий квадрат перед наименованием опции. Наличие птички включает опцию, отсутствие — выключает. Опции существенно влияют на вид документа.

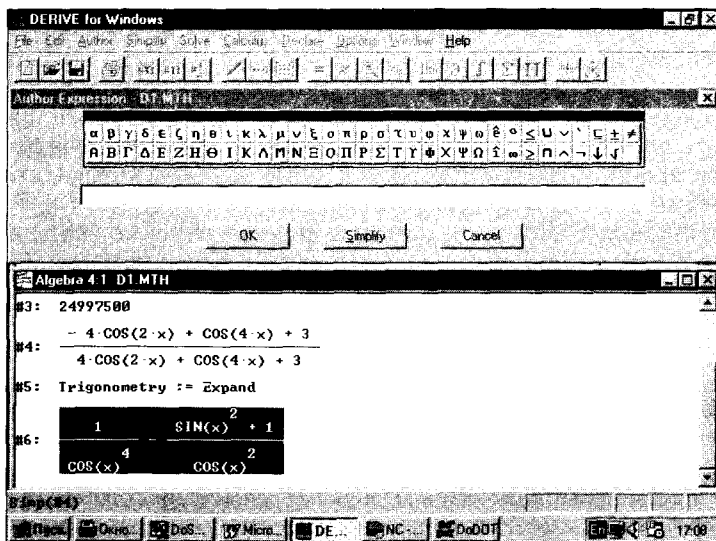
Возможна печать листов с выражениями различными шрифтами по выбору. Для этого в окне установки опций печати достаточно активизировать кнопку Fonts (фонты). Тогда появится окно выбора шрифтов, унифицированное для системы Windows. Оно также показано на рис. 4.89. С помощью этого окна можно выбрать любой шрифт из набора шрифтов, ранее установленных в систему Windows.

По умолчанию печать выполняется специальным шрифтом с именем DfW Printer, который устанавливается самой системой Derive при ее установке. Этот набор шрифтов правильно учитывает размеры символов в математических выражениях и интервалы между ними. Он содержит правильное начертание всех знаков, включая математические. Практика показывает, что при отказе от этого шрифта возникают проблемы с качеством распечатки математических выражений.

### 4.11.3. Включение/выключение диалога (The Entry Dialog)

Эта последняя команда в подменю позиции Options главного меню включает или выключает режим диалога при вводе. Такой диалог включается, если против имени команды установлена птичка, и выключается, если ее нет. При включенном диалоге автоматически устанавливается панель ввода или коррекции выражений, а под ней располагается окно выражений с линейками прокрутки (см. рис. 4.90).

**Рис. 4.90**  
Работа в режиме включенного диалога при вводе

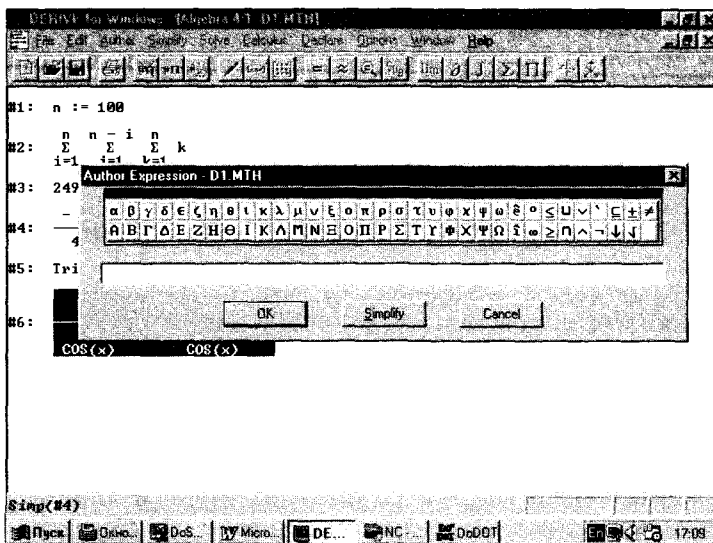


В этом режиме можно в процессе ввода или коррекции выражения просматривать любую часть документа, что нередко остро необходимо.

При отключении этой опции окно ввода и коррекции выражений просто налагается на текст документа, и последний уже нельзя просматривать. Это иллюстрирует рис. 4.91.

Заметим, что в обоих случаях окно ввода или коррекции выражения можно перемещать в любое место экрана, располагая его так, как это удобно для работы.

**Рис. 4.91**  
Работа в режиме  
выключенного  
диалога  
при вводе



## 4.12. Задание и размещение окон (Window)

В версиях Derive под MS-DOS предусматривалась разбивка окон их делением по горизонтали или по вертикали. Но при этом положение окон и их размеры были фиксированными. В новой версии под Windows возможности окон значительно расширены. В частности, можно создавать произвольное число окон с выражениями и графиками. Окна могут перемещаться и изменяться в размерах, а также располагаться по-разному. В подменю позиции Window (Окно) главного меню содержатся основные команды управления окнами и элементами пользовательского интерфейса:

- New Algebra View** — создание дополнительного окна с выражениями;
- New 2D-plot Window** — создание нового окна для двумерной графики;
- New 3D-plot Window** — создание нового окна для трехмерной графики;
- Cascade (Ctrl + Shift + C)** — каскадное расположение окон;
- Tile Horisontally (Ctrl + Shift + H)** — расположение окон по горизонтали;
- Tile Vertically (Ctrl + Shift + V)** — расположение окон по вертикали;
- Arrange Icons** — упорядочение расположения ярлыков свернутых окон;
- Toolbar** — ввод или устранение панели инструментов;
- Status Bar** — ввод или устранение строки состояния.

Рассмотрим возможности задания и вывода окон.

### 4.12.1. Создание дополнительного окна с выражениями (New Algebra View)

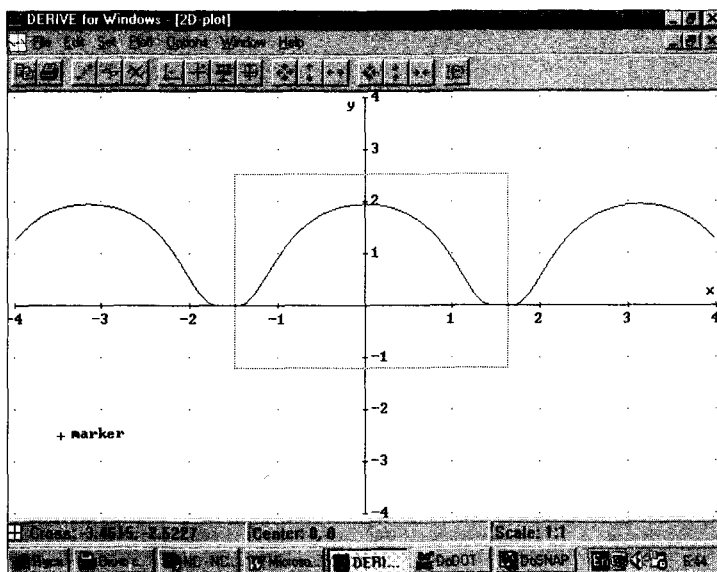
Иногда выражений в документе так много, что их не видно в одном окне. Можно открыть новое (или дополнительное) окно с выражениями, используя команду New Algebra View. Вопреки буквальному переводу этой команды, открытое окно не является новым. В нем будут видны те же выражения, что и в исходном (начальном) окне. Изображение экрана с двумя или тремя окнами,

полученными с помощью этой команды, приводилось уже не раз (например, на рис. 4.73 и 4.74). Поэтому описывать эту команду более подробно нет смысла. Отметим лишь, что как новые окна, так и старые можно изменять в размерах и перетаскивать по экрану мышью.

## 4.12.2. Создание нового окна для двумерной графики (New 2D-plot Window)

Команда New 2D-plot Window создает новое графическое окно для построения двумерных графиков (см. рис. 4.92). Такое окно содержит свою панель инструментов и свои кнопки для свертывания окна в ярлык, управления размерами и закрытия. Содержит оно и свое главное меню, несколько отличающееся по набору позиций и команд от главного меню окна выражений.

Рис. 4.92  
Окно двумерной  
графики



Основными объектами окна двумерной графики являются: строка заголовка, строка главного меню, панель инструментов, сам график, графический курсор (в виде креста +), координатная система, область выделения части графика (показана пунктирными линиями) и строка состояния. В последней указаны координаты курсора (Cross), координаты точки пересечения координатных осей Center (центром считается центр окна с графиком) и масштабные множители (Scale) по горизонтальной и вертикальной осям.

Графический курсор может перемещаться клавишами управления курсора. Перемещения будут плавными и медленными. Для быстрых, но резких перемещений надо нажимать клавиши управления курсором одновременно с клавишей Ctrl. Курсор служит для приближенного определения координат характерных точек графика и может иметь различные режимы перемещения. Они будут описаны чуть позже.

### 4.12.3. Панель инструментов окна двумерной графики

Панель инструментов окна двумерной графики дает эффективный и быстрый способ построения и модификации графиков. При этом практически не требуется обращения к командам главного меню, хотя их набор несколько больше, чем набор команд, доступных на панели инструментов.

Панель инструментов содержит шесть групп кнопок быстрого управления построением графиков. Перечислим их, начиная с первой (слева) и до последней.

#### Копирование и печать графика:

- Copy plot window** — копирование графика в буфер обмена;  
**Print** — вывод графика на принтер.

#### Управление построением графика:

- Create annotation** — создание аннотации, располагаемой по месту нахождения графического курсора;  
**Plot expression** — построение графика выделенного выражения;  
**Delete last plot** — стирание последнего построенного графика.

#### Центрирование и масштабирование графика:

- Center on cross** — центрирование графика относительно курсора;  
**Center on origin** — центрирование графика по самому себе;  
**Set scale** — установка масштаба;  
**Set range with box** — развертывание выделенного графика во все окно.

#### Расширение области графика:

- Zoom both out (F10)** — расширение области графика в обе стороны;  
**Zoom vertical out (F8)** — расширение области графика по вертикали;  
**Zoom horizontal out (F10)** — расширение области графика по горизонтали.

#### Сжатие области графика:

- Zoom both 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.

#### Выход в окно выражений:

- Algebra window (Ctrl + 1)** — возврат в окно выражений.

Действие большинства этих команд настолько очевидно, что едва ли есть смысл особо его пояснять. Хотя надо отметить, что расширение области графика означает увеличение цены деления масштабной сетки, что фактически сжимает собственно график. Аналогично сжатие области графика расширяет собственно график.

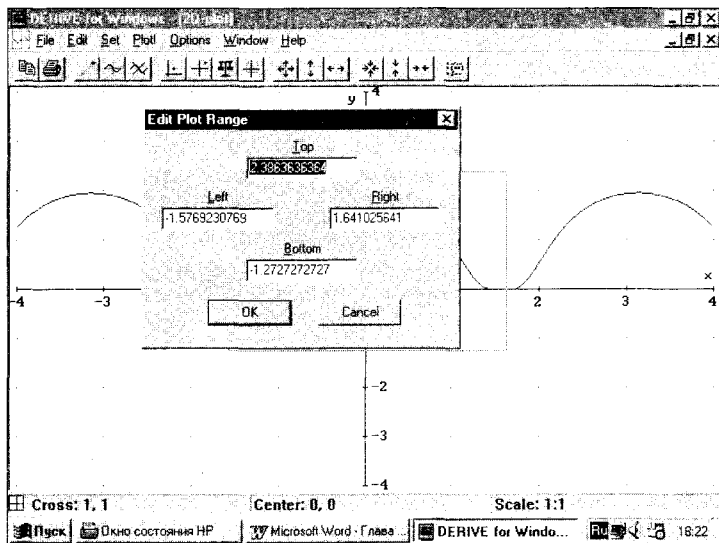
Достаточно пару минут поэкспериментировать с центрированием и масштабированием любого графика, как работа с панелью инструментов станет вам ясна. Остановимся лишь на одной интересной возможности — выделении части изображения кнопкой **Set range with box**. После нажатия этой кнопки для выделения достаточно держать нажатой левую клавишу мыши и начать ее перемещение. Появится расширяющийся прямоугольник из пунктирных линий, которым надо охватить интересующую вас часть графика. Отпустив

кнопку мыши, вы увидите появление окошка, контролирующего размеры выделенного фрагмента рисунка (см. рис. 4.93).

Уточнение размера выделенной части рисунка полезно для получения удобных значений чисел у отметок координатных осей. Если этого не сделать, у таких отметок будут проставляться довольно длинные и неудобные для применения числа, как это показано на рис. 4.94.

Рис. 4.93

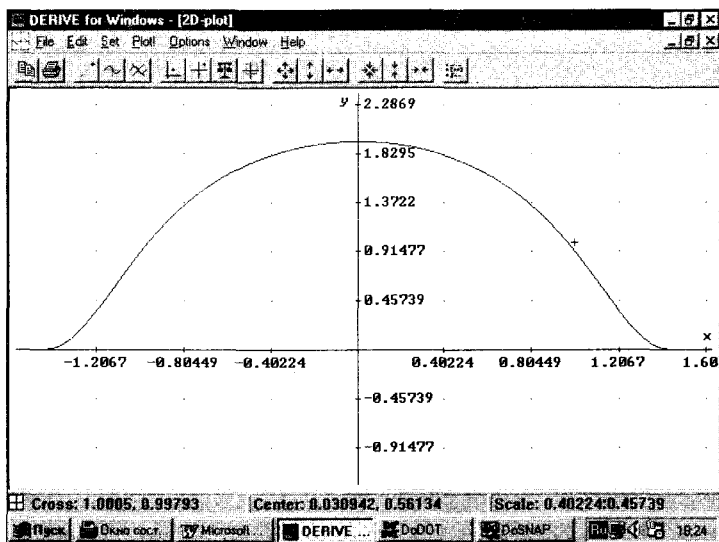
Уточнение размера выделенного фрагмента рисунка



Фактически указанная возможность реализует графическую “лупу”, позволяющую просматривать небольшие фрагменты графиков. Это полезно для выявления и изучения их особенностей.

Рис. 4.94

График выделенного фрагмента



#### 4.12.4. Главное меню окна двумерной графики

Работа с кнопками быстрого управления довольно очевидна. Тем не менее окно двумерной графики содержит еще и свое главное меню, в котором имеются следующие позиции:

- File** — закрытие окна, предварительный просмотр графиков и печать их на принтере, выход из графического окна (для файлов графиков используется формат AscSpin);
- Edit** — создание и уничтожение аннотации, стирание последнего графика и копирование графика целиком или выделенной его части в буфер обмена;
- Set** — центрирование графика, установка масштаба и размера графиков;
- Plot** — запуск построения графика выделенного выражения;
- Options** — установка различных опций (см. ниже);
- Windows** — управление окнами;
- Help** — вывод оперативной справки по Derive.

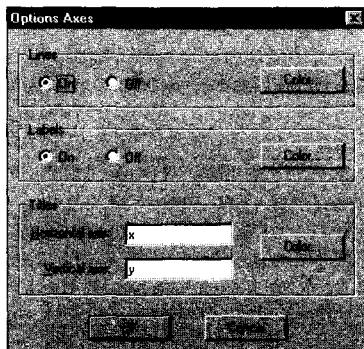
Команды главного меню, за исключением имеющих в позиции Options, уже были описаны. Поэтому ограничимся перечнем команд, вводимых в подменю позиции Options:

- Axes...** — установка параметров координатных осей и надписей у них;
- Cross...** — установка параметров графического курсора (его наличия и цвета);
- Grids...** — установка параметров масштабной сетки графика;
- Coordinate System...** — установка координатной системы (прямоугольных Rectangular и полярных Polar координат);
- Point...** — установка опции точек;
- Plot color...** — установка цвета линий графиков;
- Background color...** — установка цвета фона графиков;
- Printing...** — установка опций печати;
- Follow Mode** — установка режима свободного перемещения графического курсора;
- Trace Mode** — установка режима перемещения курсора по графику;
- Autoscal Mode** — установка автоперемещения курсора.

Большинство установок в этой позиции главного меню понятны. Некоторые, например установки цвета, аналогичны уже описанным. Поэтому остановимся на тех, которые достаточно специфичны.

**Рис. 4.95**

Окно установки параметров координатных осей



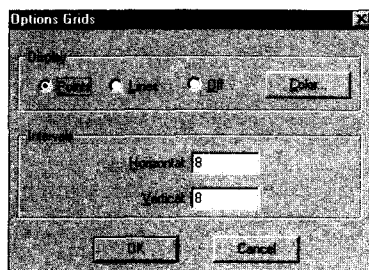
Установка параметров координатных осей создает окно, показанное на рис. 4.95.

Здесь можно задать вывод (On) или сокрытие (Off) координатных осей (Lines) и меток (Labels) на них. Можно задать также надписи (Titles) у горизонтальной (Horizontal axis) и у вертикальной (Vertical axis) осей. По умолчанию эти надписи представлены обозначениями осей — буквами  $x$  и  $y$ . Для всех элементов можно вывести окно установки цвета, используя кнопки Color.

Еще одна опция, которую стоит указать, — Grid. Она устанавливает параметры отображения масштабной сетки. По умолчанию эта сетка представлена точками в местах пересечения линий сетки. Команда Grid выводит окно, позволяющее изменить параметры масштабной сетки (см. рис. 4.96).

Рис. 4.96

Окно установки параметров масштабной сетки



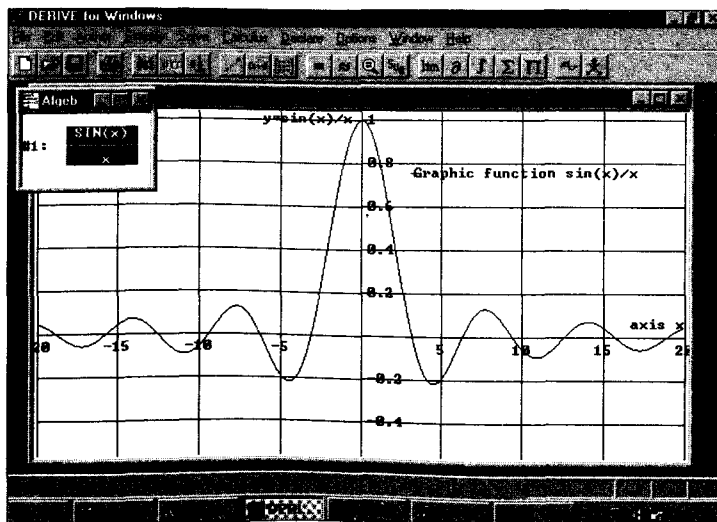
Это окно позволяет установить видимость (Display) масштабной сетки в виде точек (Point) и линий (Lines) или отсутствие сетки (Off). Кнопка Color выводит окно установки цвета точек или линий масштабной сетки.

Рис. 4.97 показывает график функции  $\sin(x)/x$  при использовании масштабной сетки из линий, выводе надписей по координатным осям и при выводе аннотации в виде надписи “Graphic function  $\sin(x)/x$ ” внутри рисунка.

В целом графики Derive с параметрами, установленными по умолчанию, выглядят вполне привычно. Но, как видно из рис. 4.97, немного поработав над

Рис. 4.97

Пример рисунка в стандартном оформлении



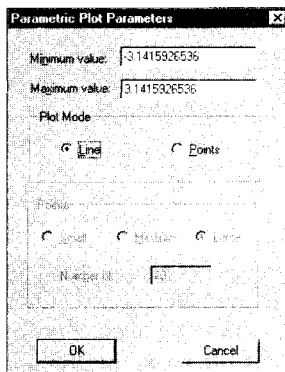


ними, можно придать им более изысканный вид. Впрочем, стоит помнить, что "на вкус и цвет товарищей нет". Так что стройте графики такого вида, который вам нравится. Главное, что бы они были аккуратными (это Derive обеспечивает) и наглядными. А вот последнее зависит уже от вас!

#### 4.12.5. Графики параметрически заданных функций

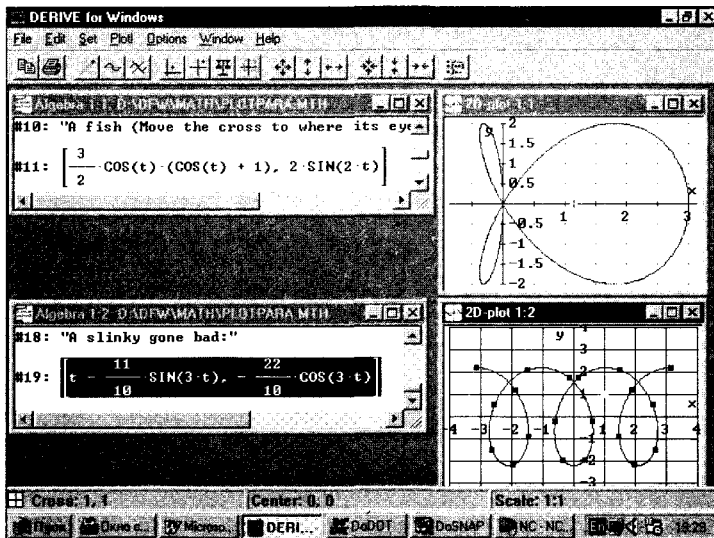
В главе 1 уже отмечалось, что Derive строит графики параметрически заданных функций. Такие функции должны быть попарно записаны в виде векторов, т. е. их надо поместить в квадратные скобки и отделить запятой. Derive сама распознает такие функции и строит их график. Предварительно она выводит на экран окно установки параметров графиков параметрически заданных функций (см. рис. 4.98).

**Рис. 4.98**  
Окно задания параметров параметрически заданных функций



На рис. 4.99 показано построение графиков для двух параметрически заданных функций. Слева выведены окна выражений с заданными в них функциями, а справа — окна с графиками этих функций.

**Рис. 4.99**  
Примеры построения графиков параметрически заданных функций

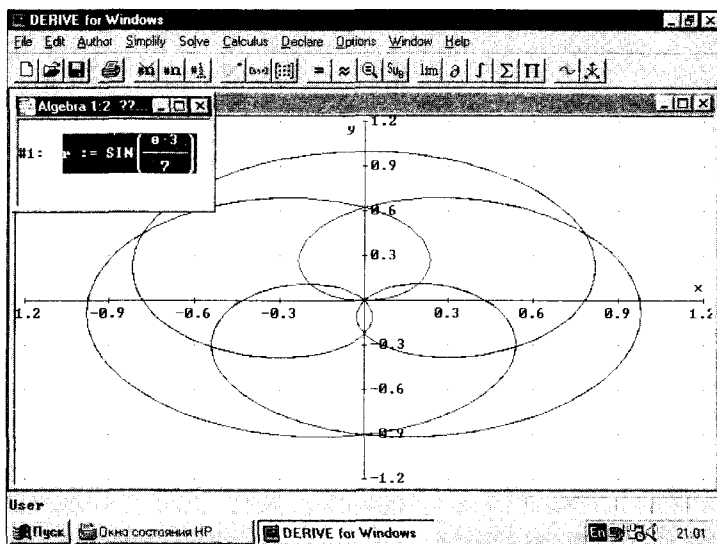


Обратите внимание на график для второй из функций. Он построен одновременно линией и точками. Для второго построения, естественно, надо было сменить установки на построение графика точками, а не линиями.

#### 4.12.6. Построение графиков функций в полярной системе координат

При построении графика в полярной системе координат надо задать функцию  $r(\theta)$  и открыть окно двумерной графики. Кроме того, надо в опции Coordinate System установить полярную систему координат (Polar). При запуске построения графика откроется окно Parametric plot parameter, в котором можно уточнить параметры построения (если надо). Нажатие кнопки ОК включает построение графика (см. рис. 4.100).

**Рис. 4.100**  
Пример построения графика в полярной системе координат



Назначение кнопок панели инструментов и команд главного меню здесь такое же, как для двумерных графиков.

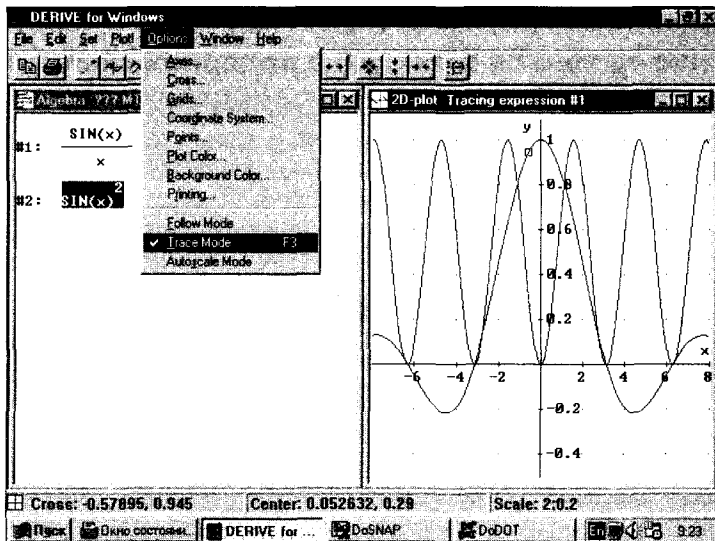
#### 4.12.7. Работа с графическим курсором

Графический курсор — эффективное средство для управления построением графиков и анализа особенностей графиков различных функций. Даже простое применение курсора для примерного определения характерных точек графика весьма полезно. В Derive под Windows введено три режима перемещения графического курсора. Режим Follow Mode особенно удобен, если надо посмотреть график функции за пределами видимого окна. В этом режиме при попытке вывести курсор за пределы графика область просмотра (или окно просмотра) скачком смещается в ту строчку, в какую сместился курсор. В режиме Autoscale Mode таких скачков нет.

Но особенно интересен режим Trace Mode, при нем графический курсор превращается из крестика в маленький квадратик и устанавливается на кривую графика, соответствующую выделенному выражению (см. рис. 4.101).

Мышью или клавишами он перемещается всегда точно по линии графика. Это повышает точность анализа особых точек кривых.

**Рис. 4.101**  
Применение режима Trace Mode для анализа кривой с помощью графического курсора



Перемещение курсора мышью, к сожалению, нельзя сделать плавным, поскольку фиксация положения курсора производится в момент щелчка левой клавишей мыши. Однако клавишами управления курсором можно обеспечить плавное движение курсора (при нажатой клавише Ctrl оно ускоряется).

Графический курсор удобно использовать также для центровки графика, для этого достаточно поместить его в точку графика, которую желательно сделать центром, и нажать кнопку центрирования, и график тут же будет перестроен.

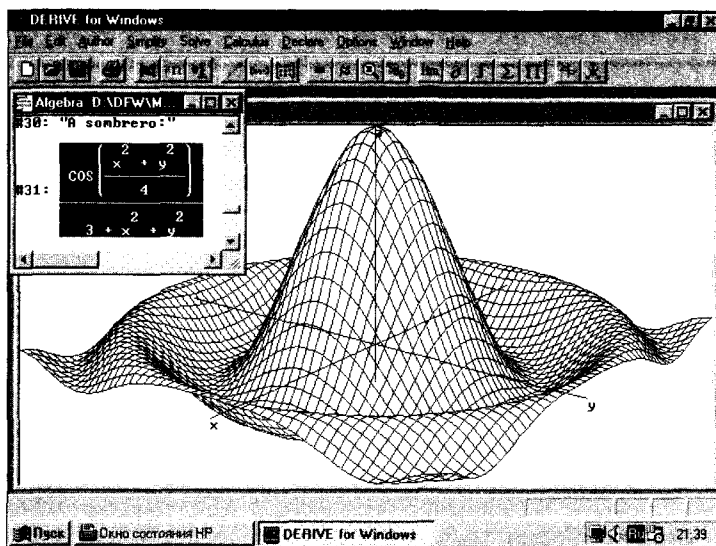
#### 4.12.8. Создание нового окна для трехмерной графики (New 3D-plot Window)

Для построения графиков 3D-поверхностей надо задать функцию двух переменных вида  $z(x, y) = f(x, y)$  или просто  $f(x, y)$ . После выделения строки с такой функцией надо использовать команду New 3D-plot. Она создает новое окно для трехмерной графики со своим главным меню и панелью инструментов. Эту же команду можно исполнить, нажав на панели инструментов окна выражений кнопку с изображением трехмерной системы координат.

После вывода окна графика надо на его панели инструментов нажать аналогичную кнопку — и график можно строить. Для наглядности можно рядом или поверх него вывести окно с исходным выражением. Осуществив все эти операции, можно получить график, представленный на рис. 4.102.

Нетрудно заметить, что график имеет вид проволочного каркаса, но его невидимые линии устранены алгоритмом удаления таких линий. Этот алгоритм включен по умолчанию, поскольку обеспечивает более наглядный вид по сравнению с графиками со всеми видимыми линиями. При необходимости, однако, алгоритм удаления невидимых линий может быть отключен.

**Рис. 4.102**  
Пример построения графика трехмерной поверхности

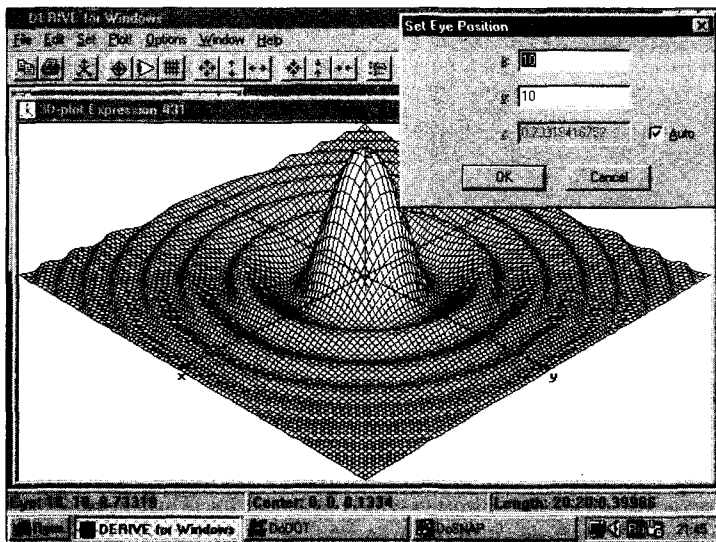


#### 4.12.9. Панель инструментов окна графиков (3D-plot)

С помощью команд панели инструментов или главного меню вид графика можно существенно изменить. В качестве примера на рис. 4.103 показан тот же график, но с другим масштабом и под другим углом зрения. Для изменения угла зрения использована кнопка панели инструментов с изображением глаза.

На рис. 4.103 видны главное меню и панель инструментов окна построения трехмерных графиков. Панель инструментов окна содержит указанные ниже кнопки (перечислены слева направо).

**Рис. 4.103**  
График трехмерной поверхности с измененными масштабом и углом зрения



**Копирование и печать графика:**

- Copy plot window** — копирование графика в буфер обмена;  
**Print** — печать графика на принтере.

**Управление построением графика:**

- Plot expression** — построение графика выделенного выражения.

**Центрирование и просмотр графика:**

- Set box center** — центрирование графика;  
**Set viewpoint** — перестройка графика под другим углом зрения;  
**Set grid panel** — перестройка графика с другим числом линий сетки.

**Расширение области графика:**

- Zoom both out (F10)** — расширение области графика в обе стороны;  
**Zoom vertical out (F8)** — расширение области графика по вертикали;  
**Zoom horizontal out (F10)** — расширение области графика по горизонтали.

**Сжатие области графика:**

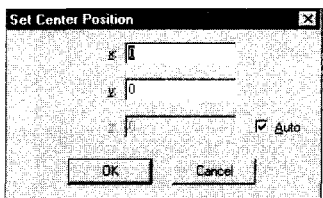
- Zoom both in (F9)** — сжатие области графика в обе стороны;  
**Zoom vertical in (F7)** — сжатие области графика по вертикали;  
**Zoom horizontal in (F5)** — сжатие области графика по горизонтали.

**Выход в окно выражений:**

- Algebra window (Ctrl + 1)** — возврат в окно выражений.

Лишь команды третьей группы (центрирование и просмотр графика) отличаются от команд окна двумерной графики. Команда **Set box center** позволяет установить координаты  $x$  и  $y$  центра графика. В режиме **Auto** координата  $z$  вычисляется автоматически, а при отказе от этого режима ее также надо указать в окне этой команды (рис. 4.104).

**Рис. 4.104**  
Окно ввода координат центра трехмерных графиков



Команда **Set viewpoint** выводит на экран окно, показанное в верхнем правом углу графика на рис. 4.103. Оно аналогично окну на рис. 4.104, и в нем нужно установить координаты точки, с которой ведется обзор трехмерной поверхности. Для изменения числа линий (grids) трехмерной поверхности по осям  $x$  и  $y$  служит команда **Set grid panel**. В ее простеньком окне вводятся два числа: количество линий по оси  $x$  и по оси  $y$ . По умолчанию они взяты равными 20, что дает быстрое построение поверхностей, но сами поверхности выглядят грубовато. Число линий порядка 50 обычно является приемлемым компромиссом между скоростью построения и наглядностью графиков трехмерных поверхностей.

### 4.12.10. Главное меню окна трехмерной графики

- Окно трехмерной графики содержит главное меню со следующими позициями:
- File** — закрытие окна, предварительный просмотр графиков и печать их на принтере, выход из графического окна;
  - Edit** — создание и уничтожение аннотации, стирание последнего графика и копирование графика целиком или выделенной его части в буфер обмена;
  - Set** — центрирование графика, установка масштаба и размера графиков;
  - Plot!** — запуск построения графика выделенного выражения;
  - Options** — установка различных опций (см. ниже);
  - Windows** — управление окнами;
  - Help** — вывод оперативной справки по Derive.

Большинство команд главного меню подобно уже описанным для окна двумерных графиков. Поэтому ограничимся описанием только тех команд, которые отличаются от известных вам. Позиция **Set** главного меню трехмерных графиков содержит подменю со следующими командами:

- Center... (Ctrl + T)** — установка координат центра;
- Eye... (Ctrl + E)** — установка координат точки обзора;
- Grids... (Ctrl + G)** — установка числа линий по осям  $x$  и  $y$ ;
- Length... (Ctrl + L)** — установка интервалов между линиями.

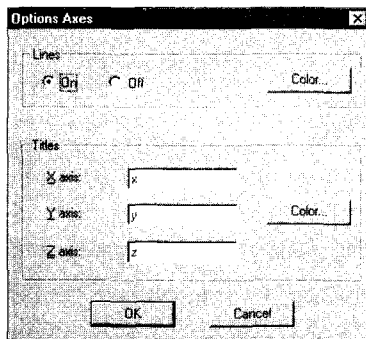
Эти установки (за исключением последней) были уже описаны при перечислении кнопок панели инструментов. Все они выводят простые окна, в которых могут устанавливаться соответствующие параметры.

В подменю позиции **Options** содержатся следующие команды:

- Axes...** — установка параметров координатных осей и надписей у них;
- Plot color** — установка цвета линий графиков;
- Background color...** — установка цвета фона графиков;
- Printing...** — установка опций печати;
- Remove Hidden Line** — устранение невидимых линий.

**Рис. 4.105**

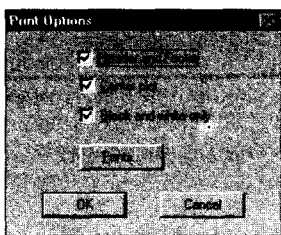
Окно установки осей **Axes** трехмерных графиков



Команда **Axes** выводит на экран окно, показанное на рис. 4.105. В этом окне можно включить или выключить вывод координатных осей трехмерных графиков и задать надписи у них. По умолчанию заданы просто наименования осей  $x$ ,  $y$  и  $z$ .

Команда **Printing** выводит окно установки опций печати трехмерных графиков (см. рис. 4.106). Это окно позволяет включить или выключить печать:

**Рис. 4.106**  
Окно опций печати



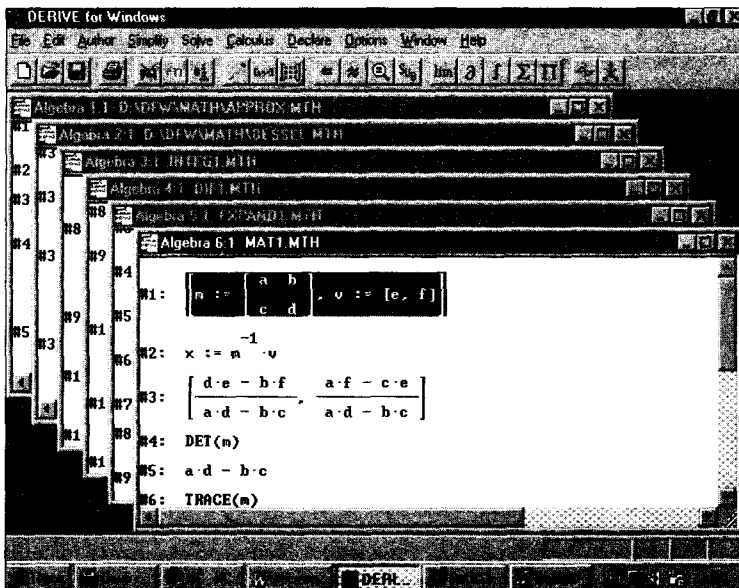
Header and Footer — нижний и верхний колонтитулы, Center plot — центральная точка графика и Black and white only — печать только черным цветом (кроме струйных принтеров, большинство доступных печатающих устройств поддерживает только одноцветную печать). Кнопка Fonts выводит панель со шрифтами для выбора подходящего набора шрифтов (об этом уже говорилось).

Остальные команды главного меню уже описывались, и на них не стоит останавливаться.

#### 4.12.11. Каскадное расположение окон (Cascade)

Мы уже не раз пользовались возможностью одновременного вывода нескольких окон. Derive допускает загрузку многих документов сразу и открытия нескольких графических окон. Если окна не полностью открыты, часть из них можно видеть на экране. Команда Cascade обеспечивает каскадное расположение окон (см. рис. 4.107).

**Рис. 4.107**  
Каскадное расположение окон

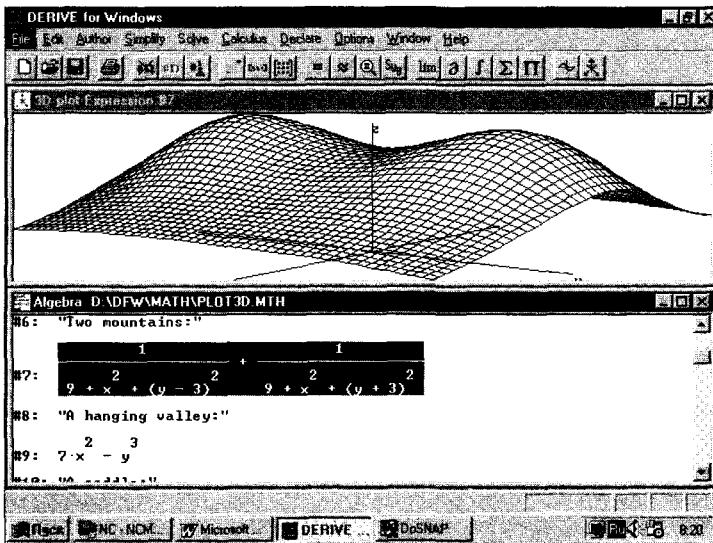


Каскадное расположение окон напоминает расположение стопки карточек. Видно верхнее активное окно и строки заголовков других окон. Щелчком мыши по ним можно сделать активным любое окно.

#### 4.12.12. Расположение окон по горизонтали (Tile Horizontally)

Возможно такое расположение окон, при котором каждому окну отводится определенное место на экране. Команда Tile Horizontally разделяет экран по горизонтали надвое и располагает в нем окна фронтально — так, чтобы в отдельном месте виднелась часть окна (с левым верхним углом). Такое деление показано на рис. 4.108.

**Рис. 4.108**  
Расположение  
окон по  
горизонтали



Если размеры окна по вертикали оказываются слишком малыми, экран делится пополам по вертикали.

#### 4.12.13. Расположение окон по вертикали (Tile Vertically)

При расположении окон по вертикали (см. рис. 4.109) пространство под окна обычно разделяется так, что все окна имеют одинаковые размеры и их форма становится близка к квадрату.

Если окна становятся слишком узкими, происходит деление окна по горизонтали. При этом окна становятся квадратными и равными по площади.

#### 4.12.14. Приведение в порядок расположения ярлыков (Arrange Icons)

Если свернуть (но не закрыть) окна, они будут представлены узкими ярлыками, которые, как правило, оказываются разбросанными по главному окну Derive (см. рис. 4.110). Каждый ярлык можно перетаскивать в любое место окна, так что беспорядок в их расположении возникает довольно быстро.

С помощью команды Arrange Icons ярлыки (их также именуют иконками) можно расположить в строгом порядке внизу основного окна Derive (см. рис. 4.111).

Обратите внимание, что каждый ярлык фактически является укороченной строкой заголовка соответствующего окна. Она содержит кнопки управ-



Рис. 4.109  
Расположение  
окон по  
вертикали

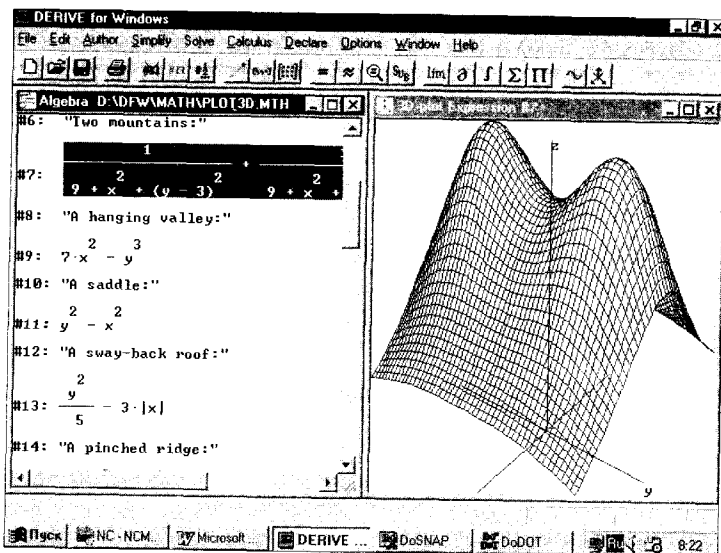
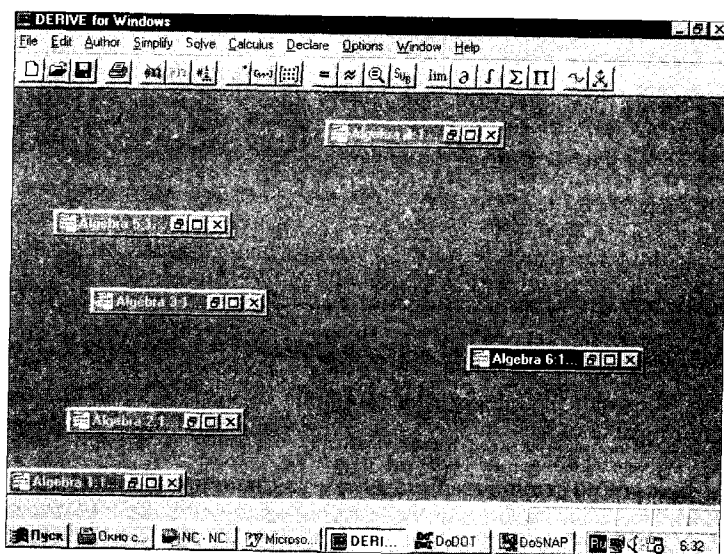


Рис. 4.110  
Ярлыки  
свернутых окон,  
расположенные  
в беспорядке

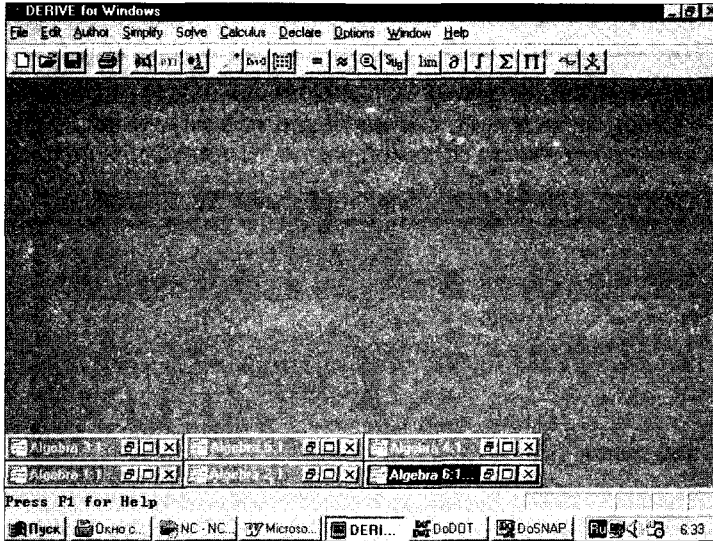


ления и титульную надпись. Бирка активного окна выделена цветом фона. Активизируя среднюю из трех кнопок в конце каждого ярлыка, можно вернуть ее в окно.

#### 4.12.15. Ввод или устранение панели инструментов (Toolbar)

Команда **Toolbar** управляет выводом панели инструментов. Она работает как переключатель. Если против названия команды установить мышью знак птички, панель инструментов возникнет на экране и ею можно будет пользоваться. В противном случае панель инструментов будет отсутствовать.

**Рис. 4.111**  
Расположение ярлыков после исполнения команды Arrange Icons



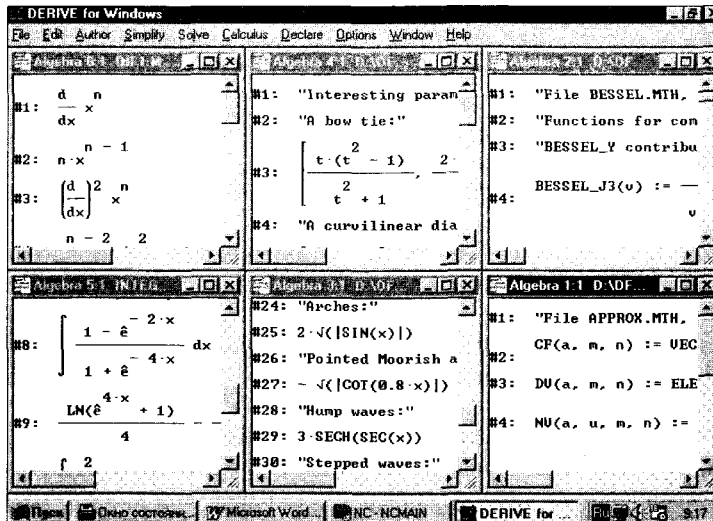
#### 4.12.16. Ввод или устранение панели состояния (Status Bar)

Команда Status Bar служит для вывода на экран или для устранения строки состояния — строки внизу основного экрана. Если перед именем этой команды установить знак птички, строка состояния возникнет на экране, иначе — будет отсутствовать.

Отключение панели инструментов и строки состояния полезно для увеличения размеров окон выражений или графики до максимально возможных размеров (см. рис. 4.112).

Панель инструментов и строку состояния можно отключать для фиксации окончательных результатов вычислений.

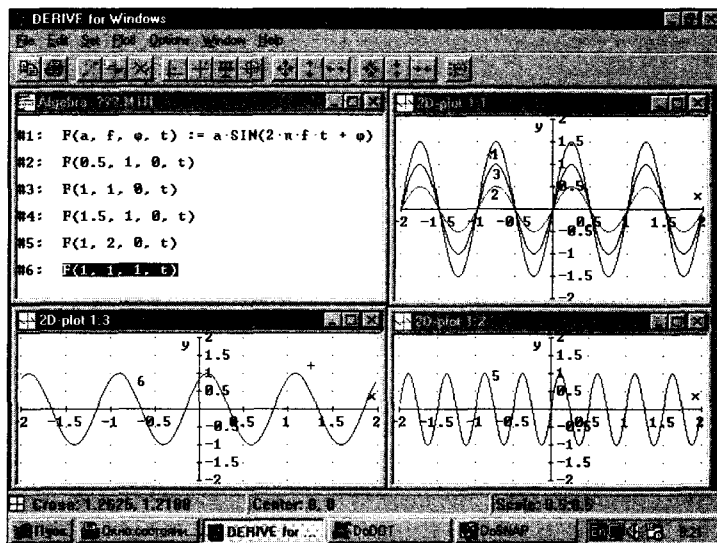
**Рис. 4.112**  
Вид экрана Derive без панели инструментов и строки состояния



### 4.12.17. Графика как средство визуализации математических понятий

Графика Derive, несмотря на ее скромные возможности (отсутствие многих специальных видов графиков, функциональной окраски и т. д.), может служить мощным средством визуализации математических и физических понятий. Это делает Derive системой весьма полезной для подготовки и решения

**Рис. 4.113**  
Представление  
графика  
синусоиды



учебных задач в школах, средних и высших технических заведениях. При этом визуализация достигается чрезвычайно просто, и любой учащийся, освоив основы работы с Derive, может в считанные минуты решать любые учебные задачи. Здесь мы приведем несколько примеров такой визуализации.

Понятие о графическом представлении функций занимает весьма важное место в курсах математики и физики. Рис. 4.113 показывает, как это понятие иллюстрируется визуальными графическими средствами Derive.

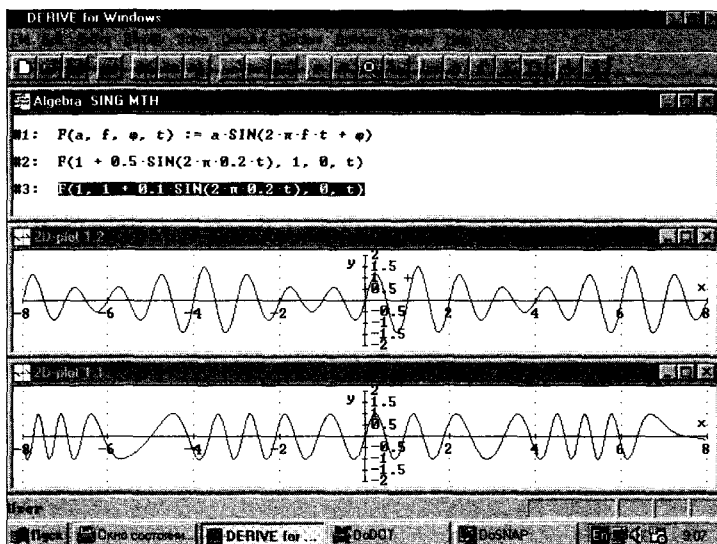
Рис. 4.113 дает представление об изменении вида синусоиды при изменении таких ее параметров, как амплитуда, фаза и частота. Особенно ценно то, что ученик может в считанные секунды изменить любой параметр и увидеть, как это скажется на графике функции.

На рис. 4.114 наглядно представлены три вида модуляции высокочастотного синусоидального колебания более низкочастотным. Это позволяет понять смысл модуляции и усвоить относящиеся к ней представления.

Большие трудности вызывает наглядное представление сложных трехмерных поверхностей, особенно если они описываются многочленными выражениями. Derive позволяет легко визуализировать такие выражения, строя графики трехмерных поверхностей для их составных частей или для полного выражения.

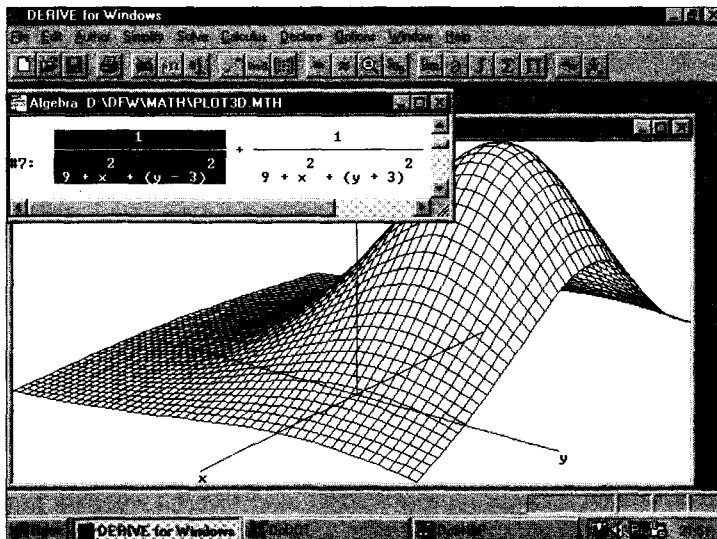
На рис. 4.115 показано некоторое выражение — функция двух переменных, имеющая два члена. Здесь строится только поверхность, описанная одним выделенным членом. Поверхность имеет вид холма, смещенного от центра графика.

**Рис. 4.114**  
Различные виды  
модуляции  
синусоидальных  
колебаний



Выделив второй член выражения, можно построить соответствующую ему поверхность. Она показана на рис. 4.116 и также представляет собой холм, смещенный относительно холма, показанного на рис. 4.115.

**Рис. 4.115**  
Первая часть  
трехмерной по-  
верхности

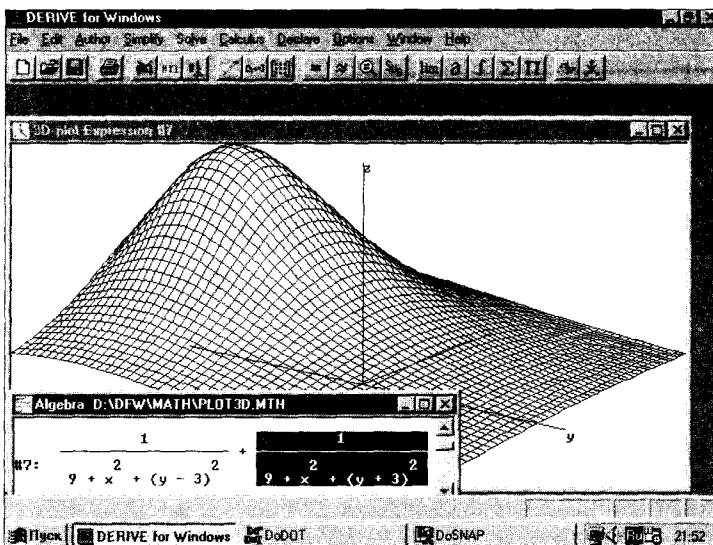


Наконец, на рис. 4.117 показан общий вид трехмерной поверхности, взятой нами для примера. Как мы видим, эта поверхность состоит из двух холмов, плавно переходящих друг в друга. Языком математики можно сказать, что она является линейной комбинацией двух поверхностей, каждая из которых описывается своим выражением.

Можно привести массу и других примеров наглядной визуализации математических понятий. Прекрасный пример — разложение функций в ряд Тей-

Рис. 4.116

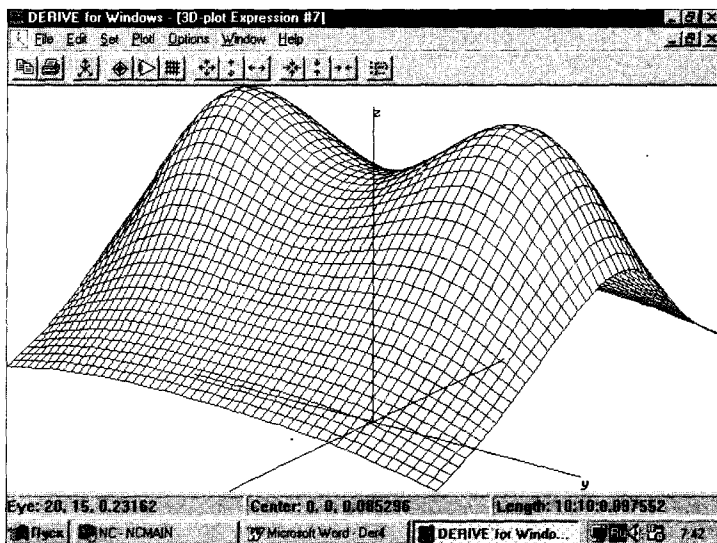
Вторая часть  
трехмерной  
поверхности



лора (см. рис. 4.68). Множество примеров визуализации аналитических вычислений было описано выше. Комбинация их с графическим выводом информации может существенно повысить наглядность представлений, которые должны усваивать учащиеся.

Рис. 4.117

Общий вид трехмерной поверхности типа "два холма"



К сожалению, Derive сама по себе предлагает весьма скромные возможности для фиксации результатов визуализации. Большой недостаток — невозможность ввода русскоязычных комментариев. Но, как уже отмечалось, эта проблема решается элементарно просто — объединением работы Derive и подходящего текстового редактора с передачей данных в него от Derive через буфер обмена.

Невозможно в Derive и сохранение главного окна с находящимися в нем различными окнами — с выражениями и с графиками. Однако и эта проблема легко решается при применении буфера обмена. Достаточно нажать клавишу Print Screen, как в буфере будет сохранен полный вид экрана. Его легко перенести в любой пакет, поддерживающий импорт графики из буфера обмена, — WordPAD, Word, Paint, PhotoShop, DoDot и пр. К тому же с помощью такого пакета можно представить результаты работы в одном из популярных форматов графических файлов, например PCX, GIF, BMP и пр.

## 4.13. Вызов справки о системе (Help)

Отличительной чертой новой версии Derive под Windows стала мощная справочная система, удовлетворяющая всем стандартам таких систем для Windows-приложений. Она выполнена с учетом требований к справочным системам под новую операционную систему Windows 95. Хотя для русскоязычных читателей ценность этой справочной системы, выполненной на английском языке, несколько ниже, она по-прежнему представляет большой интерес. Связано это, в частности, с детальной информацией о каждой команде или функции Derive, в том числе и относящейся к их синтаксису и многочисленным примерам применения.

Большинство примеров в нашей книге оригинальны и не заимствованы из справочной системы и ее фирменного описания. Поэтому примеры, приведенные в справочной системе, могут существенно пополнить знания о Derive, особенно относящиеся к отдельным частным задачам. Чем больше примеров доступно пользователю, тем легче происходит освоение системы.

Для работы со справочной системой Derive служит позиция главного меню Help (справка). Она выводит на экран подменю со следующими командами:

- Contents** — вызов контекстной справки;
- Index** — вызов индексного каталога;
- Using Help** — вызов справки о самой справочной системе;
- About Derive** — вывод информационного окна с краткими сведениями о системе.

Рассмотрим кратко работу со справочной системой.

### 4.13.1. Вызов контекстной справки (Contents)

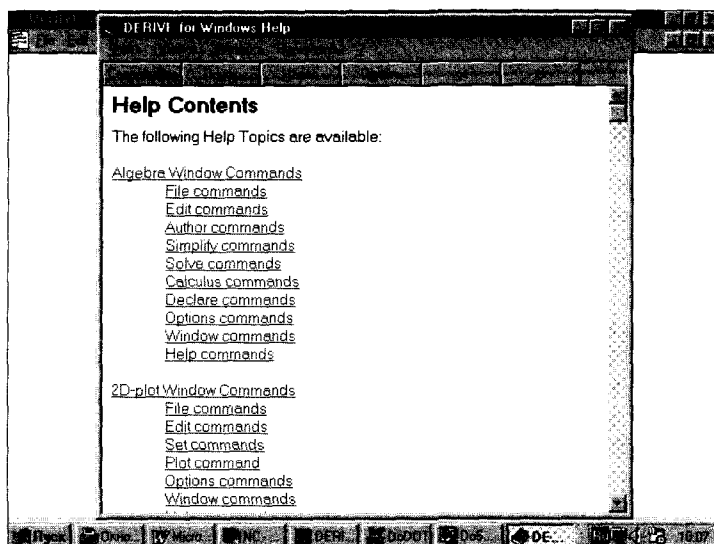
Вызов контекстной справки напоминает описание Derive в этой главе. При исполнении команды Contents на экран выводится окно со списком тем, показанное на рис. 4.118.

Выбрав какую-либо тему, можно получить перечень справок по ней и т. д., пока не будет найдена справка по заданной теме.

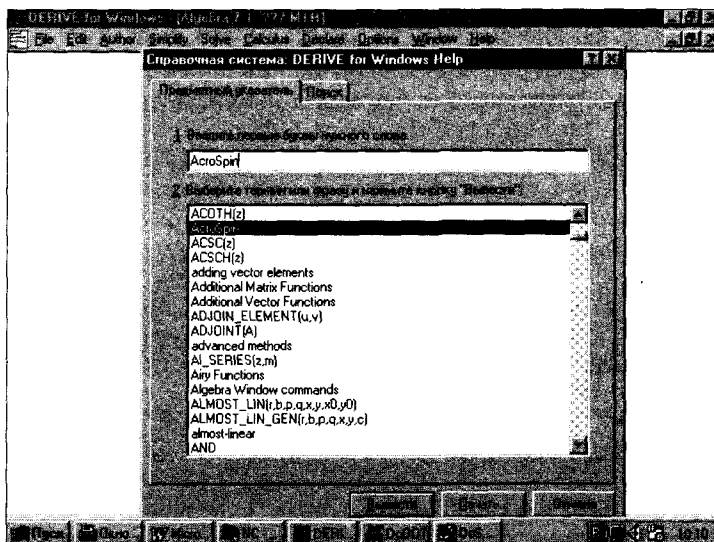
### 4.13.2. Вызов индексного каталога (Index)

Вызов индексного каталога — пожалуй, самая мощная команда справочной системы. В этом случае выводится окно справки с алфавитным каталогом, показанное на рис. 4.119. Вверху окна помещена панель ввода темы. По мере ввода наименования темы происходит ее поиск, так что порой вполне достаточно ввести лишь несколько букв названия интересующей вас темы.

**Рис. 4.118**  
Перечень тем  
контекстной  
справки



**Рис. 4.119**  
Алфавитный  
(индексный) ката-  
лог справочной  
системы Derive



Обнаружив нужную тему, достаточно щелкнуть по ней курсором мыши (название темы выделяется) и нажать клавишу Enter. Появится окно с информацией по выбранной теме.

В качестве примера на рис. 4.120 показано окно с информацией о программе AcroSpin, форматы графических файлов которой Derive использует при записи в файлы содержимого графических окон. Из этого пояснения (естественно, сделанного на английском языке) читатели могут понять, что речь идет о программе, позволяющей обрабатывать и записывать графические данные, включая графики трехмерных функций.

Рис. 4.120

Пример справки  
о программе  
AcroSpin,  
с которой  
работает Derive

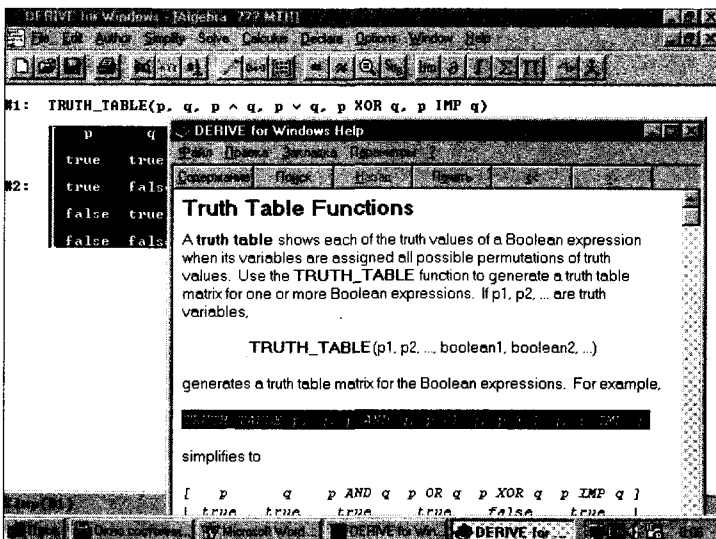


Надо отметить, что в последнее время подход к построению справочных систем существенно изменился. Раньше, в частности в предыдущих версиях Derive, справочные системы были жалкой копией справочной информации о всей системе. Подробное описание систем имелось в технической документации и в справочниках, напечатанных обычным способом.

Сейчас ситуация изменилась. Современные справочные системы с индексным и контекстным каталогами, как правило, по объему информации превосходят обычные справочники и технические руководства. Так что последние стали, по существу, сжатым изложением материалов, размещенных в весьма подробных справочных системах.

Рис. 4.121

Справка о новой  
функции Derive





Увы, для русскоязычных читателей подавляющая часть информации из справочной системы просто недоступна. Кроме того, такой системой (в отличие от книги) можно пользоваться только имея под рукой компьютер. Поэтому мощные справочные системы (даже если они написаны на русском языке) мирно сосуществуют с обычной литературой. Да и задачи у них разные: по книгам проще основательно изучать систему в спокойной обстановке, а встроенные справочники удобны для быстрого поиска и уточнения отдельных возможностей системы в процессе прямой работы с ней.

Хотя наша книга носит справочный характер и содержит довольно полное описание системы Derive, автор не гарантирует, что в ней описаны все без исключения свойства этой системы. Поэтому не исключено, что в редких случаях обращение к справочной системе Derive может оказаться вполне уместным (см. рис. 4.121).

### 4.13.3. Вызов справки о самой справочной системе (Using Help)

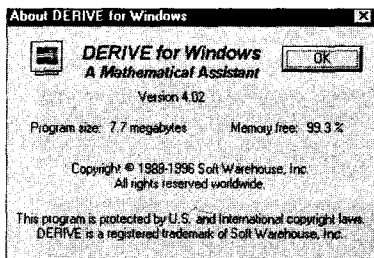
Для начинающих пользователей работа с самой справочной системой представляет определенные трудности. Поэтому команда Using Help выводит на экран сведения о том, как работать со справочной системой. Поскольку все это уже описано, да и вряд ли нашего читателя заинтересуют детали описания на английском языке, на этом разговор о справках здесь можно закончить.

### 4.13.4. Краткая информация о системе (About Derive)

Последняя команда About Derive из подменю позиции Help главного меню выводит краткую информацию о системе Derive в виде окна, показанного на рис. 4.122.

Рис. 4.122

Окно  
с данными  
о системе  
Derive



Посмотрев на экран, вы можете узнать, что система была разработана в 1996 г. (но выпущена на рынок практически только в 1997 г.), что это версия 4.02 под Windows, что она занимает на диске 7,7 Мбайта памяти и что она сейчас использует очень мало памяти (указан процент свободной памяти).

## Заключение

Итак, мы закончили знакомство с новейшими версиями популярной математической системы Derive. При своих довольно скромных размерах она является эффективной системой компьютерной алгебры и при решении ряда задач превосходит многие другие системы по скорости счета и надежности решений. Функциональные возможности системы тщательно взвешены, в них нет почти ничего лишнего. В этом ее достоинство.

Оценивая систему как жемчужину символьной математики, нельзя не отметить и некоторые ее недостатки.

- По современным понятиям интерфейс версии системы 3.11 под MS-DOS примитивен, в нем даже не предусмотрена работа с мышью (этот недостаток начисто исключен в версии 4.02 под Windows).
- Не предусмотрена запись сессии со всеми установками окон выражений и графики, так что каждый раз их приходится повторять заново.
- Графические возможности не позволяют осуществлять функциональную закрашку поверхностей и не обладают некоторыми современными возможностями научной графики (линии равного уровня, векторная графика, графики пересекающихся поверхностей и т. д.).
- Текст и графика не совмещаются в пределах одного окна.
- Многие специальные функции работают лишь в виде интегральных представлений, которые появляются при преобразованиях этих функций (например, дифференцировании) вместо самих функций с измененным порядком.

Надо полагать, что в последующих версиях под Windows эти недостатки будут хотя бы частично устранены, что повысит конкурентоспособность системы и продвинет ее в направлении решения более сложных задач. Однако в целом Derive — система, претендующая на лидерство среди математических систем начального уровня.

При умелом обращении Derive будет полезна для решения повседневных научных, инженерных и экономических задач, но главное назначение системы — учебное. Derive может занять все еще пустующую нишу в нашем образовании, где значение роли математических систем лишь начинает осознаваться.

# Приложение 1. Перевод сообщений Derive

Эти сообщения были включены в файл derive.lan прежних версий Derive и служили для перевода англоязычных сообщений оригинальной версии системы на русский язык. Их полезно знать при работе с новыми англоязычными версиями. Поэтому ниже воспроизводится перевод этих сообщений.

|                                        |                                                                     |
|----------------------------------------|---------------------------------------------------------------------|
| Abandon expressions                    | — Покинуть выражения                                                |
| Aborted                                | — Работа прервана                                                   |
| Active                                 | — Активно                                                           |
| Adapter                                | — Адаптер                                                           |
| A Mathematical Assistant               | — Математический помощник                                           |
| Amount                                 | — Количество                                                        |
| Angle sums & multiple angles           | — Суммы углов и кратные углы                                        |
| At column                              | — В столбце                                                         |
| At line                                | — В строке                                                          |
| Auto                                   | — Авто                                                              |
| Auto change color                      | — Автоматическая смена цвета                                        |
| Auto z coordinate                      | — Автоматическая установка z-координаты                             |
| Auto z length                          | — Автоматическая установка<br>длины по оси z                        |
| Axis                                   | — Ось                                                               |
| Before                                 | — Перед                                                             |
| Bottom                                 | — Низ                                                               |
| Bounds                                 | — Границы                                                           |
| Cannot do implicit plots               | — Не могу рисовать неявные функции                                  |
| Cannot find COMMAND.COM                | — Не могу найти COMMAND.COM                                         |
| Case                                   | — Выбор                                                             |
| Change                                 | — Сменить                                                           |
| Color                                  | — Цвет                                                              |
| Columns                                | — Столбцы                                                           |
| Compute time:                          | — Время счета:                                                      |
| Coordinates                            | — Координаты                                                        |
| Degree                                 | — Степень                                                           |
| Digits                                 | — Цифры                                                             |
| Dimension                              | — Размер                                                            |
| Direction                              | — Направление                                                       |
| Disk full; delete some files and retry | — Диск переполнен; удалите несколько<br>файлов и попробуйте еще раз |
| Display                                | — Дисплей                                                           |
| Display axes                           | — Показывать оси                                                    |
| Drive not ready                        | — Устройство не готово                                              |
| Dubious accuracy                       | — Сомнительная точность                                             |
| End                                    | — Конец                                                             |
| Enter DOS command                      | — Введите команду DOS                                               |
| Enter base between 2 and 36            | — Введите основание между 2 и 36                                    |
| Enter bound on solution                | — Введите границы поиска решения                                    |
| Enter color number:                    | — Введите номер цвета:                                              |
| Enter column number                    | — Введите номер столбца                                             |

- Enter columns per tick mark** — Введите число столбцов на деление
- Enter coordinates** — Введите координаты
- Enter cross coordinates** — Введите координаты пересечения
- Enter expansion point** — Введите точку разложения
- Enter expression** — Введите выражение
- Enter expression or press Enter** — Введите выражение или нажмите Enter
- Enter file name** — Введите имя файла
- Enter grid panels** — Введите число ячеек решетки
- Enter interval length** — Введите длину интервала
- Enter label number** — Введите номер выражения
- Enter left bound** — Введите левую границу
- Enter limit point** — Введите точку предела
- Enter line length** — Введите длину строки
- Enter line number** — Введите номер строки
- Enter margin columns** — Введите поле в столбцах
- Enter margin lines** — Введите поле в строках
- Enter matrix element (** — Введите матричный элемент (
- Enter maximum degree** — Введите максимальную степень
- Enter mode number** — Введите номер режима
- Enter name** — Введите имя
- Enter name or type "default"** — Введите имя или введите "default" (по умолчанию)
- Enter number of columns** — Введите число столбцов
- Enter number of elements** — Введите число элементов
- Enter number of points** — Введите число точек
- Enter number of rows** — Введите число строк
- Enter numerical output style** — Введите формат вывода чисел
- Enter option** — Введите пункт меню
- Enter page columns** — Введите количество столбцов на странице
- Enter page lines** — Введите количество строк на странице
- Enter parameter domain** — Введите область изменения параметра
- Enter path (e.g. C:\DERIVE)** — Введите доступ (например, C:\DERIVE)
- Enter plot accuracy (0 to 9)** — Введите тщательность рисования (от 0 до 9)
- Enter replacement for** — Введите замену для
- Enter replacement for subexpression** — Введите замену для подвыражения
- Enter right bound** — Введите правую границу
- Enter rows per tick mark** — Введите количество строк на деление
- Enter significant digits** — Введите количество значащих цифр
- Enter significant digits to display** — Введите количество выводимых значащих цифр
- Enter units per tick mark** — Введите шкалу (число единиц на деление)
- Enter variable** — Введите переменную
- Enter variable or press Enter** — Введите переменную или нажмите Enter
- Enter variables in desired order** — Введите переменные в выбранной вами последовательности
- Enter vector element** — Введите элемент вектора
- Enter window number** — Введите номер окна
- Enter window type** — Введите тип окна
- Evaluating row** — Вычисляю ряд

|                                         |                                                    |
|-----------------------------------------|----------------------------------------------------|
| <b>Exponential transformations</b>      | — Экспоненциальные преобразования                  |
| <b>Expression too large to edit</b>     | — Выражение слишком велико для редактирования      |
| <b>File not found</b>                   | — Файл не найден                                   |
| <b>Font</b>                             | — Фонт                                             |
| <b>From</b>                             | — С                                                |
| <b>Input</b>                            | — Ввод                                             |
| <b>Left</b>                             | — Левый                                            |
| <b>Length</b>                           | — Длина                                            |
| <b>Lines</b>                            | — Строк                                            |
| <b>Loading expression #</b>             | — Загружается выражение #                          |
| <b>Logarithm transformations</b>        | — Логарифмические преобразования                   |
| <b>Lower limit</b>                      | — Нижний предел                                    |
| <b>Max</b>                              | — Максимум                                         |
| <b>Min</b>                              | — Минимум                                          |
| <b>Mode</b>                             | — Режим                                            |
| <b>Monitor</b>                          | — Монитор                                          |
| <b>Must be in graphics mode</b>         | — Необходим графический режим                      |
| <b>Mute warning messages</b>            | — Звуковой предупреждающий сигнал                  |
| <b>Next plot</b>                        | — Следующий график                                 |
| <b>No finite real samples</b>           | — Нет конечных действительных точек                |
| <b>No solutions found</b>               | — Решений не найдено                               |
| <b>Operator</b>                         | — Оператор                                         |
| <b>Order</b>                            | — Порядок                                          |
| <b>Output</b>                           | — Вывод                                            |
| <b>Overwrite existing file</b>          | — Записать поверх существующего файла              |
| <b>Paper</b>                            | — Бумага                                           |
| <b>Plotting bottom of expression</b>    | — Строится нижняя часть поверхности                |
| <b>Plotting expression</b>              | — Строится выражение                               |
| <b>Point</b>                            | — Точка                                            |
| <b>Points</b>                           | — Точки                                            |
| <b>Press H for help</b>                 | — Для получения справки нажмите латинскую букву H  |
| <b>Press any key to continue</b>        | — Для продолжения нажмите любую клавишу            |
| <b>Press letter for desired subject</b> | — Нажмите букву, соответствующую выбранному пункту |
| <b>Printer</b>                          | — Принтер                                          |
| <b>Printer not ready</b>                | — Принтер не готов                                 |
| <b>Printing expression #</b>            | — Печатается выражение #                           |
| <b>Printing window...</b>               | — Печатается окно...                               |
| <b>Projecting row</b>                   | — Проектируется ряд                                |
| <b>Questionable solution</b>            | — Сомнительное решение                             |
| <b>Quit printing expressions</b>        | — Прервать печать выражений                        |
| <b>R — Return to Derive</b>             | — R — Возврат в Derive                             |
| <b>Range</b>                            | — Диапазон                                         |
| <b>Reading file...</b>                  | — Считывается файл...                              |
| <b>Remove hidden lines</b>              | — Убрать невидимые линии                           |
| <b>Resolution (или Reso)</b>            | — Разрешение                                       |
| <b>Return for all or select 1:</b>      | — Возврат для всех или выберите 1:                 |

|                                               |                                                   |
|-----------------------------------------------|---------------------------------------------------|
| <b>Return for no more or select next:</b>     | — Возврат для прекращения или выберите следующий: |
| <b>Right</b>                                  | — Правый                                          |
| <b>Rows</b>                                   | — Строки                                          |
| <b>Saving expression #</b>                    | — Записывается выражение #                        |
| <b>Saving file...</b>                         | — Пишу в файле...                                 |
| <b>Scaling</b>                                | — Масштабирование                                 |
| <b>Select amount of factoring</b>             | — Выберите размер факторизации                    |
| <b>Select approach direction</b>              | — Выберите направление подхода                    |
| <b>Select arithmetic mode</b>                 | — Выберите тип арифметики                         |
| <b>Select arrow key mode</b>                  | — Выберите режим клавиш-стрелок                   |
| <b>Select background</b>                      | — Выберите фон                                    |
| <b>Select case mode</b>                       | — Выберите учет регистра                          |
| <b>Select character set</b>                   | — Выберите набор символов                         |
| <b>Select coordinate system</b>               | — Выберите систему координат                      |
| <b>Select display adapter</b>                 | — Выберите адаптер дисплея                        |
| <b>Select display mode</b>                    | — Выберите режим дисплея                          |
| <b>Select input mode</b>                      | — Выберите режим ввода                            |
| <b>Select monitor type</b>                    | — Выберите тип монитора                           |
| <b>Select operator</b>                        | — Выберите операцию                               |
| <b>Select orientation</b>                     | — Выберите ориентацию                             |
| <b>Select plot mode</b>                       | — Выберите режим построения                       |
| <b>Select plot point mode</b>                 | — Выберите режим печати точек                     |
| <b>Select plot point size</b>                 | — Выберите размер точек                           |
| <b>Select preferred branch for roots</b>      | — Выберите желаемую ветвь корней                  |
| <b>Select print range</b>                     | — Выберите диапазон печати                        |
| <b>Select printer type</b>                    | — Выберите тип принтера                           |
| <b>Select resolution</b>                      | — Выберите разрешение                             |
| <b>Select save range</b>                      | — Выберите блок сохраняемых выражений             |
| <b>Select size</b>                            | — Выберите размер                                 |
| <b>Select text size</b>                       | — Выберите размер текста                          |
| <b>Select value or domain of</b>              | — Выберите значение или диапазон                  |
| <b>Select zoom axis</b>                       | — Выберите изменение масштаба осей                |
| <b>Select zoom direction</b>                  | — Выберите направление изменения масштаба         |
| <b>Set</b>                                    | — Установить                                      |
| <b>Simplifying expression</b>                 | — Упрощается выражение                            |
| <b>Size</b>                                   | — Размер                                          |
| <b>Start</b>                                  | — Начало                                          |
| <b>Style</b>                                  | — Стиль                                           |
| <b>Syntax error detected at cursor</b>        | — Над курсором найдена синтаксическая ошибка      |
| <b>Text</b>                                   | — Текст                                           |
| <b>This version not designed for HP 95LX.</b> | — Эта версия не подходит для HP 95LX.             |
| <b>Too few real sample points in box</b>      | — В области слишком мало действительных точек     |
| <b>Too many variables</b>                     | — Слишком много переменных                        |
| <b>Top</b>                                    | — Вершина                                         |

|                            |                                                       |
|----------------------------|-------------------------------------------------------|
| Toward                     | — К                                                   |
| Trig power transformations | — Преобразования тригонометрических степеней          |
| Type                       | — Тип                                                 |
| Type EXIT to return to     | — Напечатайте EXIT и нажмите Enter, чтобы вернуться в |
| Upper                      | — Верхний                                             |
| Upper limit                | — Верхний предел                                      |
| Use                        | — Используйте                                         |
| Width                      | — Ширина                                              |
| Window                     | — Окно                                                |

## Приложение 2. Интерпретация записи примеров для Derive 3.11

| Знаки Derive                    | Математические знаки | Пояснение         |
|---------------------------------|----------------------|-------------------|
| ----                            | —                    | Черта деления     |
| { }                             | ( )                  | Круглые скобки    |
| inf                             | $\infty$             | Бесконечность     |
| $x^2$                           | $x^2$                | Квадрат $x$       |
| SQRT                            |                      | Квадратный корень |
| $\int$                          | $\int$               | Знак интеграла    |
| $\left( \frac{d}{dx} \right)^2$ | $\frac{d^2}{dx^2}$   | Производная       |

# Литература

1. **Дьяконов В. П.** *Справочник по расчетам на микрокалькуляторах*. Изд. 3-е, перераб. и доп. М.: Наука, 1989. 464 с.
2. **Дьяконов В. П.** *Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ*. М.: Наука, 1987. 240 с.
3. **Дьяконов В. П.** *Форт-системы программирования персональных ЭВМ*. М.: Наука, 1992. 352 с.
4. **Дьяконов В. П.** *Компьютер в быту*. Смоленск: Русич, 1996. 640 с.
5. **Дьяконов В. П.** *Популярная энциклопедия мультимедиа*. М.: АБФ, 1996. 416 с.
6. **Дьяконов В. П.** *Справочник по применению системы Eureka*. М.: Наука, 1993. 96 с.
7. **Дьяконов В. П.** *Mercury — отличная система для всех* // Монитор-Аспект. 1995. № 5. С. 86.
8. **Дьяконов В. П.** *Как выбрать математическую систему?* // Монитор-Аспект. 1993. № 2. С. 22.
9. **Дьяконов В. П., Пеньков А. А.** *Современные математические системы* // PC Week/RE. 1996. № 43 (67). С. 42.
10. **Дьяконов В. П.** *Расширяемые системы для численных расчетов MatLAB* // Монитор-Аспект. 1993. № 2. С. 26.
11. **Дьяконов В. П.** *Справочник по применению системы PC MatLAB*. М.: Наука, 1993. 112 с.
12. **Дьяконов В. П.** *Система MathCAD/Справочник*. М.: Радио и связь, 1993. 128 с.
13. **Дьяконов В. П.** *MathCAD прорывается в Windows* // Монитор-Аспект. 1993. № 2. С. 42.
14. **Дьяконов В. П.** *MathCAD — пять с плюсом!* // Монитор-Аспект. 1995. № 2. С. 109.
15. **Очков В. Ф.** *MathCAD PLUS 6.0 для студентов и инженеров*. М.: Компьютер Press, 1996. 238 с.
16. *MathCAD 6.0 PLUS. Финансовые, инженерные и научные расчеты в среде Windows 95.* / Пер. с англ. М.: Филинь. 1996. 712 с.



17. **Дьяконов В. П.** *MathCAD PLUS 6.0 — ключ к тайнам математики* // Домашний компьютер. 1997. № 4. С. 37.
18. **Дьяконов В. П.** *Справочник по применению системы Derive*: М.: Наука, 1996. 144 с.
19. **Дьяконов В. П.** *Derive — жемчужина символьной математики* // Монитор-Аспект. 1993. № 2. С. 36.
20. **Дьяконов В. П., Бирюков С.** *Derive в России*. Монитор-Аспект, 1995, № 3, с. 56.
21. *Руководство пользователя DERIVE. Математический Помощник на ПК.* / Рус. пер.: МНИИТЦ “Скан” — Soft Warehouse, Inc. 1993. 348 с.
22. **Дьяконов В. П.** *Maple V — мощь и интеллект компьютерной алгебры!* // Монитор-Аспект. 1993. № 2. С. 48.
23. **Говорухин В. Н., Цибулин В. Г.** *Введение в Maple V. Математический пакет для всех*. М.: Мир, 1997. 208 с.
24. **Дьяконов В. П.** *Mathematica 2.0 под MS-DOS и под Windows* // Монитор-Аспект. 1993. № 2. С. 52.
25. **Дьяконов В. П.** *Mathematica 2.1 для Windows: от слов к делу!* // Монитор-Аспект. 1995. № 4. С. 125.
26. **Дьяконов В. П.** *Mathematica 2.2.2 — на пути к совершенству* // Монитор-Аспект. 1996. № 6. С. 96.
27. **Аладьев В. З., Шишаков М. Л.** *Введение в среду пакета Mathematica 2.2*. М.: Филинь, 1997. 368 с.
28. **Дэвенпорт Дж., Сирэ И., Турнье Э.** *Компьютерная алгебра. Системы и алгоритмы алгебраических вычислений*. М.: Мир, 1991. 352 с.
29. **Гантмахер Ф.** *Теория матриц*. М.: Наука, 1988. 552 с.
30. *Справочник по специальным функциям с формулами, графиками и математическими таблицами*. Под ред. Абрамовица М. и Стиган И. М.: Наука, 1979. 832 с.
31. **Корн Г., Корн Т.** *Справочник по математике для научных работников и инженеров*. М.: Наука, 1973. 832 с.
32. **Воднев В. Т., Наумович А. Ф., Наумович Н. Ф.** *Основные математические формулы*. Минск: Вышэйшая школа, 1988. 270 с.

33. **Хювейн Э., Сеппянен И.** *Мир Лисна*. В 2 томах. М.: Мир, 1990. Том 1 — 447 с. Том 2. 319 с.
34. **Саймон Барри.** *Символьная математика: новые времена — новые формы* // PC Magazine/RE. 1992. № 5.
35. **Berry J.S., Graham E., Watkins A.J. (University of Plymouth).** *Learning Mathematics through DERIVE*. Ellis Horwood. 1993. 371 p.
36. **Bohm Josef.** *Teaching Mathematics with DERIVE*. Great Britain: Chartwell-Bratt Ltd. 1992. 300 p.
37. **Brian H. Denton.** *Learning Linear Algebra Through DERIVE*. Ellis Horwood. 1994. 300 p.
38. **Rich A.D., Stoutemyer D.R.** *Inside the DERIVE Computer Algebra System* // England: The International Derive Journal. 1994. Vol. 1. № 1. P. 3.