Enter

«СОЛОН-Р»

Владимир Дьяконов

Системы компьютерной алгебры DERIVE

Самоучитель

серия

ПОЛНОЕ

руководство пользователя

В. П. Дьяконов

СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ DERIVE

Самоучитель и руководство пользователя

Москва
СОЛОН-Р
2002

Дьяконов В. П.
Системы компьютерной алгебры Derive: Самоучитель и руководство пользователя. / Дьяконов В. П. — М.: СОЛОН-Р, 2002. — 320 с.

ISBN 5-93455-139-6

В книге описаны популярные, особенно в образовании, простые и мощные системы компьютерной математики Derive под MS-DOS и Windows 95/98/NT, Они существуют как отдельные программные системы и как системы, встроенные во флэш-память новейших графических микрокалькуляторов TI-89, TI-92 и TI-92 Plus. Впервые в нашей литературе описана новая версия системы Derive 6. Системы обеспечивают проведение как простых, так и самых сложных вычислений, как в символьном (аналитическом), так и в численном виде, имеют развитые возможности цветной двухмерной и трехмерной графики и богатые библиотеки внешних расширений. Дано множество примеров применения Derive для решения математических, научно-технических и экономических задач. Показаны возможности интеграции Derive с текстовыми процессорами класса Word. Для широкого круга специалистов в области математики, физики и прикладных наук, а также для школьников, студентов высших учебных заведений и пользователей ПК.

ISBN 5-93455-139-6

Предисловие

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

Albert Rich, Joan Rich, David Stoutemyer (Создатели Derive)

Компьютеры — от микрокалькуляторов [1] до супер-ЭВМ — 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 литературу [2,3]. И по сей день такие сферы их 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 обывательским синонимом программному обеспечению компьютеров стал термин «математика компьютеров», даже в том случае, когда это программное обеспечение имеет очень далекое от серьезной математики назначение.

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

Как же помочь большинству пользователей компьютерами в обучении современным основам символьных (аналитических) вычислений и, главное, в их повседневном и полезном применении в своей будущей работе? Эти задачи сейчас решает компьютерная алгебра — новейшее направление в информационных технологиях, получившее бурное развитие в 90-годы [13, 14] именно благодаря появлению ПК. В эти годы, в целом ряде своих обзоров и книг, автор данной книги предсказывал, что такие системы для ПК должны занять должное место в арсенале не только математика, но и инженера, научного работника, студента и даже школьника. Так оно и случилось!

Разработка систем компьютерной математики с элементами символьных (аналитических) вычислений стала высочайшим достижением компьютерного «интеллекта». Их возможностям посвящена обобщающая монография автора [5] — самая полная книга по компьютерной математике, насчитывающая 1296 страниц.

Новые системы компьютерной математики выполняют, наряду с традиционными численными расчетами, символьные (аналитические) вычисления и имеют мощные средства их графической визуализации. Уже появился и широко используется ряд систем компьютерной математики для массовых персональных компьютеров (ПК): muMath, Reduce, Mathcad, Derive, Maple, Mathematica, MATLAB и др. Эти системы постоянно обновляются и расширяются, и их возможности и корректность вычислений постоянно растут. Все системы постепенно превращаются в универсальные системы компьютерной математики.

Derive одна из хорошо известных систем такого рода. Ей принадлежит почетная роль системы, ориентированной, прежде всего, на образование, причем не только высшее, но и начальное (школьное). На это указывает полное название системы — Derive a Mathematical Assistant (Derive — математический ассистент).

Derive, пожалуй, единственная из систем компьютерной математики, которая выпускается сразу в трех вариантах — под все еще живую MS-DOS, под все версии Windows и даже в составе встроенного программного обеспечения массовых графических микрокалькуляторов фирмы Texas Instruments. Эта, поразительно компактная система подлинная жемчужина среди систем компьютерной математики, до сих пор имеет версии для работы в среде MS-DOS [6, 7]. Вспомним, что в наших школах, вузах и университетах все еще используются сотни тысяч компьютеров с процессорами от 286 до 486. Derive под MS-DOS может устанавливаться и даже работать с гибкого диска. Поразительная компактность Derive достигнута реализацией системы на языке искусственного интеллекта MuLISP, который (как и саму систему) создали специалисты небольшой фирмы Soft Warehouse Inc. (США).

Но самым примечательным явлением в развитии систем компьютерной математики стал перенос Derive в hardware (встроенное программное обеспечение) массовых микрокалькуляторов фирмы Texas Instruments TI-89/92/92 Plus. Это в конечном счете привело к вливанию фирмы Soft Warehouse Inc. в состав одной из крупнейших мировых корпораций — Texas Instruments, издавна славящейся разработками микрокалькуляторов, измерительных приборов и даже собственных микропроцессоров и больших интегральных микросхем (БИС).

Объективности ради надо отметить, что первой попыткой создать ЭВМ со встроенной системой символьных вычислений была разработка под руководством советского академика Глушкова инженерных ЭВМ класса «Мир» с языком программирования «Аналитик». Разработчики этих машин, увы, не поддержанные должным образом в бывшем СССР, могли лишь мечтать о том, что такие машины со временем превратятся в микроминиатюрные массовые микрокалькуляторы, выпускаемые с объемом во многие миллионы единиц.

Derive уже используется во многих десятках тысяч школ Запада. Эта система хорошо известна и у нас [7]. Однако литературы по ней было очень мало и она выходила с большим опозданием по сравнению с развитием самих этих систем. Так, первая книга автора по самой простой и массовой системе символьной математики Derive [8] была сдана в издательство еще в конце 1991 года, но вышла в свет лишь в 1996 году. Причиной такой «оперативности» ведущего научного издательства бывшего СССР «Наука. Физматлит», первым взявшегося за издание подобной литературы, стало разрушение системы научного книгоиздания в те годы, связанное с издержками перехода к рыночной экономике.

Однако такое важное направление его, как издание литературы по системам компьютерной математики, все же сумело выжить. А сейчас прорыв в издании литературы по системам символьной математики в образовании стал очевидным. Этот вид литературы получил признание во многих коммерческих издательствах, работающих (к их чести) быстро и оперативно.

Следом за книгами автора и его многочисленными, отмеченными выше обзорами появились отдельные книги по крупным математическим системам Mathcad, Maple, Mathematica 2 и MATLAB. Среди них и вторая книга автора по системе Derive [9]. Система Derive описана также (среди других систем) в книгах [5, 10]. Наконец, вышла книга [11] с описанием практикума по решению математических задач в высшей школе на основе применения системы Derive (к сожалению, устаревшей версии).

Но выпуск даже нескольких книг по каждой системе компьютерной математики далеко не отражает степени их практического применения и явно отстает от разработок новых версий математических систем. Так, только в по системе Derive за рубежом издано более сотни книг.

Уже после издания книг [8—10] вышло несколько новых версий Derive, в том числе в 2000 году вышла новейшая 32-разрядная версия Derive 5 под Windows 95/98/NT/2000. Данных о ней в нашей литературе вообще нет, как и по новому поколению научных калькуляторов фирмы Texas Instruments TI-89/92/92 Plus, в которых используется ядро символьной математики на основе Derive.

Данная книга хотя бы частично устраняет этот пробел, что особенно важно в связи с наметившейся тенденцией заметного роста интереса к образованию. Такая тенденция вполне понятна — наша молодежь и старшее поколение наконец-то осознали, что в условиях свертывания неконкурентоспособного производства образование позволяет им не только быть при деле в ближайшие несколько лет, но и легче адаптироваться к новым условиям жизни. Очевидно также, что только серьезное образование является единственной возможностью добиться возрождения отечественного производства, способного развиваться в условиях рыночного хозяйства, рухнувшего «железного занавеса» и открытых границ между странами. В итоге конкурс в вузы и университеты заметно возрос и возрос интерес к математическим системам.

Эта книга впервые в России (и в странах СНГ) описывает все современные версии одной из самых удачных систем компьютерной алгебры для целей образования Derive — от широко у нас известных Derive 3 и 4 под MS-DOS и Derive 4 под Windows до новейшей версию Derive 5 под Windows 95/98. Все эти системы совместимы, имеют практический одинаковый набор функций и команд и могут обмениваться своими файлами. Более того, книга вводит читателя в мир Derive, который (что особенно примечательно) уже не ограничен только версиями этой системы для персональных компьютеров. Теперь он охватывает новейшее поколение программируемых графических микрокалькуляторов фирмы Texas Instruments TI-89/92/92 Plus. Таким образом, Derive стала первой системой компьютерной математики, которая проникла в такой массовый вид изделий, как научные микрокалькуляторы.

Все описанные в данной книге версии Derive (в том числе Derive 5.0 под Windows) можно найти на получивших широкое распространение CD-ROM. Однако следует помнить, что имеющиеся на CD-ROM программы не являются лицензионными и поэтому не содержат никакой документации или руководств пользователя и не могут пользоваться поддержкой фирм-разработчиков.

Автор отнюдь не является сторонником использования получаемых таким не вполне цивилизованным образом программ и иных материалов. Хотя бы потому, что на многих дисках с прежними версиями Derive он обнаружил и свою книгу [8], хотя за разрешением опубликовать ее на CD-ROM — никто из инкогнито — создателей этих CD-ROM не обращался. Забавно, что книга была скопирована прямо с указанием об уголовной ответственности за нарушение авторских прав. То же можно сказать и о ряде других книг автора по системам компьютерной математики.

Но, как говорится, «из песни слов не выкинешь» — подавляющее большинство наших пользователей все еще пользуются именно полученными таким образом программами, в том числе и Derive. И потому нуждаются в справочниках и руководствах пользователя по их применению. Кстати, нужны они и вполне законным владельцам компьютерных математических систем, благо все они англоязычные и найти в книжных магазинах описания этих систем на русском языке просто невозможно.

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. Данная книга по стилю изложения является самоучителем по всем современным версиям и реализациям Derive — словом, по миру Derive. Она не повторяет фирменное описание этих систем и содержит множество авторских примеров их применения. Большая часть материалов книги ориентирована на пользователей любой версией системы Derive, за исключением материалов по разным интерфейсам этих систем под MS-DOS и Windows, сильно различающимся и потому описанным раздельно. По полноте описания книга может претендовать и на справочное руководство по системам класса Derive.

Состав книги

Эта книга лишь частично подготовлена по материалам ранее опубликованных справочных книг автора по системам Derive [8, 9]. При этом их материал существенно переработан и дополнен. Большая часть материала этой книги посвящена новейшей версии системы Derive 5, поступившей на рынок в 2000 году. При этом использовались результаты ее бета-тестирования автором.

Но главное — книга подготовлена в стиле учебного курса для самостоятельного освоения описанных в ней новых средств (самоучителя). Его можно рассматривать как введение в применение систем компьютерной математики с развитыми возможностями в символьных (аналитических) вычислениях. При этом полнота материала книги такова, что он может претендовать на роль практического пособия по применению систем класса Derive всех классов, руководства по их использованию и даже справочника по всем их обширным возможностям. Книга оснащена подробным оглавлением, которое можно рассматривать как тематический каталог по материалам книги. Перевод англоязычных сообщений Derive на русский язык дан в Приложении.

Автор обращает особое внимание читателей на то, что книга является цельным справочным пособием по системам класса Derive. Пользователям MS-DOS версий Derive полезно ознакомиться с уроками по Derive под Windows (и даже с материалами по калькуляторам), поскольку в них можно найти множество дополнительных примеров применения Derive. Они вынесены в эти уроки, 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 последними версиями Derive 4 и Derive 5 под Windows необходим материал уроков **1—5**, содержащих описание основополагающих концепций систем класса Derive и основ их применения. Наконец, уроки **7—12**, посвященные программированию в среде систем Derive и описанию их расширения с помощью внешних библиотек, в равной мере необходимы пользователям любой версией Derive.

Математические алгоритмы и методы вычислений в книге описаны очень кратко — их детали заинтересованный читатель может найти в специальной литературе **[14—19]**. Ввиду малой доступности зарубежной литературы ее список ограничен работами **[20—23]**. Довольно внушительный список зарубежной литературы по системам класса Derive можно найти в Интернете, в частности, на странице корпорации Texas Instruments.

Благодарности

Книга подготовлена по результатам инициативных работ кафедры физической и информационной электроники Смоленского государственного педагогического университета. Работы автора были поддержаны рядом грантов (Минобразования РФ в области фундаментального естествознания, дважды фондом Сороса (гранты соросовских профессоров по программе ISSEP) и фирмы Wolfram Research Inc.)» что существенно облегчило работу автора и позволило подготовить обширную серию книг по системам компьютерной математики, изданных в ряде издательств России. Теперь эта серия пополнилась описанием мира систем Derive.

Нельзя не высказать слова признания в адрес фирмы Soft Warehous Inc. (США) и всех ее сотрудников, создавших такую замечательную систему, как Derive a Mathematical Assistant и сумевших найти ей должное место на рынке современных математических систем. Особую благодарность автор выражает одному из главных создателей системы Derive Альберту Ричу (Albert Rich) за предоставление автору на бета-тестирование новейшей версии Derive 5 под Windows, что позволило заметно ускорить подготовку данной книги.

Автор благодарит также главу фирмы Wolfram Research Inc. (создавшей систему Mathematica) С. Вольфрама (Stepfen Wolfram) и представителя этой фирмы Олега Маричева за представленную ему возможность плодотворной работы во время длительной стажировки в США на этой фирме. Она помогла автору в работе над данной книгой в полном ее объеме и получить (через скоростной Интернет фирмы) объемные новые материалы по системам компьютерной математики, включая Derive.

Автор выражает глубокую признательность С. В. Бирюкову — представителю научного центра МНИИТЦ «Скан», успешно представляющего и даже русифицирующего системы Derive в России, за плодотворное сотрудничество и своевременно представленную информацию по новым версиям системы Derive. Он признателен и Л. Д. Лозинскому, который первым обратил его внимание на жемчужину символьной математики — систему Derive и опубликовал первые методические пособия по ней, увы, крайне ограниченным тиражом (в силу чего ссылка на них опущена).

Предупреждения и пожелания

Автор и издательство сделали все возможное, чтобы материал книги был доступен, актуален, объективен и точен. Однако компьютерная алгебра настолько новое направление, что терминология и перевод отдельных терминов в ней еще не устоялись, порою могут трактоваться двояко, неточно и даже ошибочно. Кроме того, Derive настолько мощная и весьма своеобразная система, что нет гарантии в том, что в сжатые сроки подготовки книги что-то в ее описании и не было упущено.

Автор и издательство не несут никакой ответственности за неправильное истолкование каких-либо терминов и положений, относящихся к системе Derive, и за просчеты в ее применении при решении конкретных задач пользователя. Автор и издательство не несут также ответственности за нелегальное использование системы Derive пользователями, вне зависимости от того, откуда она была инсталлирована на их ПК, а также за любой моральный и материальный вред, который может.(скорее всего, гипотетически) вытекать из работы пользователя с системой Derive.

Эти предупреждения вовсе не означают наличия каких-либо особых недостатков системы Derive и ее описания в данной книге. Их упоминание просто является долгом автора и издательства перед читателями, вытекающим из современного юридического подхода к использованию программных средств и их описаний.

В то же время автор и издательство весьма заинтересованы в оценке данной книги читателями и будут признательны всем, кто готов высказать свое мнение по материалу книги, его полноте и качеству. Отзывы и пожелания следует направлять по адресу издательства или по адресу электронной почты автора: dyak@keytown.com.

Урок 1. Знакомство с миром систем Derive

- Общая характеристика систем Derive
- Derive в калькуляторах TI-89/92/92 Plus фирмы Texas Instruments
- Возможности систем класса Derive для ПК
- Derive в образовании и в развитии
- Отражение мира Derive в Интернете

Общая характеристика систем Derive

Место Derive в семье компьютерных математических систем

Роль математики в науке, технике и в 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 граждан, кто по влечению, а кто и 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 (в буквальном переводе computer — вычислитель).

И вое же десятилетиями компьютеры помогали в решении математических задач лишь малочисленной элите научного и инженерного корпуса, занятой сложными и трудоемкими расчетами. При этом компьютеры скорее не облегчали, а усложняли решение многих повседневных математических задач. Ведь для этого 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 и безошибочного выполнения численных математических вычислений. Она могла реализоваться лишь при профессиональном уровне применения возможностей компьютеров. Одни при этом ставили задачи и предлагали математические методы их решения, другие программировали компьютеры, а третьи тестировали решения задач.

С появлением программируемых микрокалькуляторов и персональных компьютеров (ПК) вычислительные машины стали доступны рядовому инженеру или студенту [1—3]. Процесс вычислений приобрел персональный характер. Однако подход к применению компьютеров 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 Inc. или PC и AT MATLAB фирмы MathWork Inc.) реализовали возможности численных методов анализа и имели удобную для работы пользователя рабочую среду в MS-DOS.

В то же время эти системы демонстрировали поразительную беспомощность в 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)^2+cos(x)^2`, хотя результаты вычисления этих выражений очевидны даже для не слишком преуспевающего в учебе ученика средней школы. Ну а вычисление интегралов или производных в аналитическом виде могла лишь присниться в бреду!

Между тем ЭВМ и программные системы, выполняющие символьные вычисления и способные выдавать результаты в виде аналитических зависимостей, известны вообще-то давно. Лидирующая роль в разработке таких ЭВМ у нас принадлежала школе академика В.М. Глушкова, создавшей малые инженерные ЭВМ серии «Мир» с языком «Аналитик» для проведения символьных вычислений. Это были первые компьютеры (правда, не персональные), в которых символьные вычисления были реализованы на аппаратном уровне. К сожалению, что отмечено в введении, эта ветвь вычислительной техники в дальнейшем не была поддержана в должной мере в бывшем СССР, и лидерство перешло к зарубежным разработчикам таких средств.

За рубежом также был создан ряд языков программирования и программных систем для символьных операций: muMATH, REDUCE, Maple, Mathematica и др., создавших основу для развития компьютерной алгебры [13, 14]. Среди этих систем одной из самых простых и получивших массовое распространение была система muMATH, реализованная на многих мини- и микро-ЭВМ. Небольшая фирма Soft Warehouse Inc. (США) на основе этой системы в последние годы разработала универсальную математическую систему Derive a Mathematical Assistant (далее просто Derive), описанию современных версий которой и посвящена данная книга.

Derive, вне всякого сомнения, принадлежит исключительная роль — это была первая система компьютерной- алгебры, реализованная на персональных компьютерах (ПК). К тому же это самая маленькая из систем компьютерной алгебры, образно говоря, подлинная жемчужина символьной математики. Derive поистине система для **всех!** Даже последние версии Derive 3 и 4 под MS-DOS могут устанавливаться на почти «музейных» ПК класса IBM PC XT и AT без жесткого диска. Затраты по передаче такой системы по Интернету 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 класса 8086, 286, 386 и 486. А ведь до сих пор большинство наших школ и вузов вынуждено пользоваться такими компьютерами.

Да и система Derive 4 под Windows, обладая превосходным (и в то же время удивительно простым) интерфейсом приложений под Windows 95/98, занимает на жестком диске менее 8 Мбайт памяти, что во много раз меньше памяти, которую пожирают такие системы, как Mathcad 6.0, Mathematica 2 и Maple. А уж что говорить об их новейших вер'сиях, имеющих явную тенденцию к непомерному «разбуханию» — Mathematica 3/4, к примеру, занимает на жестком диске уже около 120 Мбайт, а инсталляционная версия Mathcad 2000/2001 не помещается даже в сотне Мбайт. А рекордсменом по пожиранию жесткого Диска стала мощная матричная система MATLAB — ее новые версии занимают на жестком диске до 1,5 Гбайт памяти!).

Итак, новое поколение интегрированных математических систем, начиная с системы Derive, освоило компьютерную алгебру. Так называют совокупность средств, способных выполнять аналитические преобразования и вычисления, например, вычислять неопределенные и определенные интегралы в символьном виде, разлагать функции в ряд, выполнять различные преобразования с математическими выражениями, подставлять одни выражения на места других, находить производные и пределы любых функций и т. д. и т. п.

При автоматическом выполнении таких операций широко используются рекурсивные алгоритмы, порою требующие больших объемов памяти, прежде всего из-за глубокой рекурсии, осуществляемой в ходе сложных преобразований больших математических выражений. Из-за этого первые версии Derive не могли эффективно использоваться при решении достаточно сложных математических задач.

Между тем новые 32-разрядные версии Derive XM, которые и описаны в настоящей книге, могут обращаться к расширенной памяти до 4 Гбайт, что обеспечивает построение очень развитых деревьев решений и позволяет решать сложнейшие задачи символьной математики. Однако они могут прекрасно работать и с памятью (ОЗУ) всего в 2 Мбайта. Различные версии Derive распространяются в России и странах СНГ фирмой МНИИТЦ «Скан».

Русская пословица гласит: «мал золотник, да дорог». Это вполне относится к системам Derive — как под MS-DOS, так и под Windows. Объективности ради следует отметить, что новейшая версия Derive 5 под Windows заметно «пополнела». За счет этого она имеет новый современный интерфейс и новую графику с возможностью функциональной окраски трехмерных фигур. Документы в ней готовятся в форме ноутбуков — блокнотов самого изысканного и современного вида, — причем рисунки могут помещаться прямо в документы — в предшествующих версиях они выводились только в отдельные окна. И при этом система остается намного более компактной, чем ее маститые конкуренты.

Разумеется, чудес на свете не бывает — по ряду показателей, и прежде всего обилию графических возможностей и работе со специальными математическими функциями, Derive заметно отстает от новых версий таких систем, как Mathcad, Maple, Mathematica и MATLAB. Однако даже опытные пользователи имеют очень малый «КПД» при использовании этих больших систем и порою просто не знают о большинстве их возможностей. Derive, будучи более простыми системами, используется куда эффективнее. В общем, как говаривал наш великий полководец Суворов, «пуля дура, а штык — молодец». Сказанное о штыке можно вполне отнести к Derive.

К сожалению, многие из систем компьютерной математики для персональных компьютеров (Maple V R5/6/7, Mathematica 3/4, Mathcad 2000/2001 и др.) пока очень громоздки и требуют от ПК больших аппаратных ресурсов. Их можно использовать лишь на довольно дорогих ПК с процессорами класса Pentium, с большим объемом оперативного запоминающего устройства (ОЗУ) — практически более 8 Мбайт. Но даже на таких ПК эти системы часто оказываются «тугодумами», проводя символьные операции в явно замедленном темпе, нервирующем пользователей, привыкших чуть ли не к мгновенному решению умеренно сложных численных задач на таких ПК..

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 сиотемами.

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 дискетках и CD-ROM).

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

Фактически Derive это единственная малая система компьютерной математики, 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 с системой на современных ПК с микропроцессорами Pentium, имеющими повышенную производительность. А появление систем Derive 4.* и 5.* (звездочка означает существование ряда модификаций систем) под Windows дает пользователям возможность воспользоваться всеми преимуществами пользовательского интерфейса приложений под операционные системы Windows 3.1/3.11/95/98/ME/2000.

Derive в переводе означает «извлекать» или «наследовать». Видимо, оба эти значения учитывали разработчики системы. Derive унаследовала лучшие черты от некогда самой массовой системы компьютерной алгебры — mu-MATH. Поэтому, несмотря на новизну, Derive тщательно апробированная, надежная и быстрая система. В своей обобщающей обзорной статье [23] главные создатели системы Derive и основатели фирмы Soft Warehouse Inc. Albert D. Rich и David R. Stoutemyer отмечают, что на создание системы Derive (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 неожиданные математические закономерности и результаты. В этом нет ничего удивительного — ядро Derive содержит около 1000 функций, написанных на языке MuLISP, и реализовано 23 тысячами программных строк. Для сравнения можно отметить, что куда большая система Mathematica 2.2.2 содержит в ядре, написанном на языке C++, столько же функций.

Удивительная компактность ядра Derive связана именно с реализацией на языке искусственного интеллекта MuLISP, прекрасно подходящего для реализации сложных рекурсивных функций, сложно взаимодействующих друг с другом в ходе осуществления символьных преобразований. О языке Lisp можно прочитать в книге [19]. Это один из лучших языков для создания экспертных систем, к коим, в сущности, и относятся системы класса Derive.

Среди математических систем по-своему уникальна система Mathcad. 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, или в обычных языках программирования высокого уровня вместо математических знаков вводят слова, обозначающие математические функции и операторы. Например, sqrt или sqr вместо знака квадратного корня (а иногда возведения в квадрат). Надо отметить, что большинство пользователей ПК настолько привыкли к таким обозначениям, что они не только не вызывают особого протеста, но и кажутся предпочтительными перед бесконечным тыканием курсором мышки в многочисленные панели с математическими знаками (что имеет место 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 (в версии под Windows) вводом 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 Derive 5.* под Windows.

Derive в калькуляторах TI-89/92/92 Plus фирмы Texas Instruments

Роль микрокалькуляторов

В России появление персональных компьютеров привело к заметному и ничем не оправданному падению интереса к серьезным калькуляторам. К сожалению, наши предприятия оказались неспособными выпускать микроминиатюрные микрокалькуляторы новых поколений. Да и производственное объединение «Кристалл» (основной производитель отечественных программируемых микрокалькуляторов) осталось на Украине. Выпуск специализированных микросхем для современных калькуляторов не менее сложное дело, чем производство микропроцессоров. В итоге выпуск микрокалькуляторов сошел на нет и наши пользователи вынуждены даже простые вычисления (кроме разве что самых элементарных) выполнять на персональных компьютерах, увы, все еще доступных далеко не всем и не всегда удобных в работе.

Между тем за рубежом не принято «стрелять из пушки по воробьям» — простые вычисления там по-прежнему выполняют на микрокалькуляторах. И не только простые вычисления, но и довольно сложные. Ведущие фирмы (Texas Instruments, Hewlett Packard, Casio и др.), выпускающие современные микрокалькуляторы, имеют от этого миллиардные прибыли (в долларах) и постоянно совершенствуют этот вид товаров массового спроса. Увы, но наши граждане просто не догадываются о новых и обширных возможностях этих умных машинок.

Принципиально новым видом микрокалькуляторов стали графические программируемые микрокалькуляторы, практически стирающие разницу между ними и компьютерами в выполнении любых (в том числе достаточно сложных) вычислений. Выпущены десятки миллионов графических микрокалькуляторов, и они начинают проникать на рынок России.

Как уже отмечалось, Derive стала первой из систем компьютерной математики, реализованной в виде hardware (т. е. на аппаратном уровне), в новейших микрокалькуляторах фирмы Texas Instruments. Объективности ради надо отметить, что второй такой системой, причем более мощной, чем Derive, стала система Maple V. Она аппаратно реализована в микроминиатюрных компьютерах Cassiopeia фирмы Casio. Эти компьютеры, однако, стоят намного дороже микрокалькуляторов TI-89/92 (тоже, кстати, не слишком дешевых — их стоимость составляет 150—200 долларов). Но выпуск Cassiopeia с встроенной системой Maple пока довольно ограничен.

TI-89 и **TI-92/92** Plus — микрокалькуляторы нового **поколения,** рекомендуемые к применению в школах с расширенной математической подготовкой и в высших учебных заведениях. Обе модели имеют абсолютно идентичные возможности и параметры и отличаются только внешним оформлением и размерами экономичного жидкокристаллического экрана.

Калькулятор TI-89

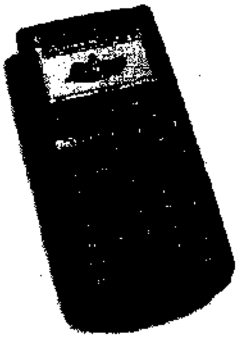


Рис. 1.1. Внешний вид научного графического микрокалькулятора TI-89 с возможностью выполнения символьных операций

Калькулятор TI-89 выполнен в стиле классического калькулятора (рис. 1.1). Экран его жидкокристаллического дисплея, с разрешением 160x100 точек (пикселей), расположен у верхней короткой стороны корпуса. Расположение клавиш характерно для микрокалькуляторов — оно удобно для ввода чисел и арифметических операторов, но неудобно для ввода текстов. В частности потому, что **клавиши** с буквами расположены в алфавитном порядке.

Калькулятор TI-89 — основная модель в ряду новых микрокалькуляторов с встроенной системой компьютерной алгебры. Его удобно держать в ладони, и он размещается в кармане пиджака.

Калькуляторы TI-92/92 Plus

Калькулятор TI-92/92 Plus выполнен в стиле микрокомпьютера (рис. 1.2). У него QWERTY клавиатура, дисплей повышенного разрешения (240x128 пикселей) и размера и даже вполне современный встроенный графический манипулятор с 8 направлениями перемещения графического маркера. Наиболее важные клавиши продублированы, например, имеются три клавиши ввода Enter в наиболее удобных местах клавиатуры.

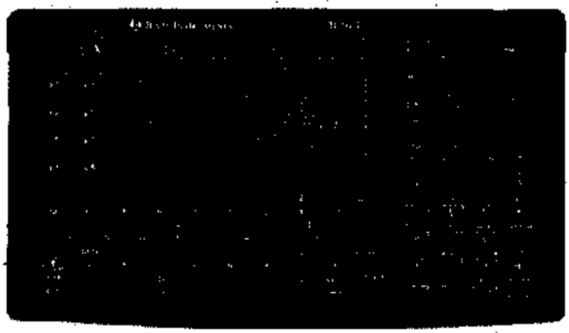


Рис. 1.2. Внешний вид калькулятора TI-92

Этот калькулятор ориентирован на школьников классов с углубленной математической подготовкой, студентов вузов и университетов, инженеров и научных работников. Фактически это специализированный на персональные (и даже личные) вычисления современный микрокомпьютер.

Возможности калькуляторов TI-89/92/92 Plus

Обе модели калькуляторов способны в командном режиме выполнять практически все встречающиеся на практике простые и умеренно сложные вычисления. Это арифметические действия, операции с действительными и комплексными числами и строками, решение систем линейных, нелинейных и даже дифференциальных уравнений (рис. 1.3), операции с векторами и матрицами, аналитические и численные вычисления производных и интегралов, вычислением пределов, нулей, минимумов и максимумов функций, разложение функций и выражений в ряд Тейлора и многие другие виды вычислений.

Калькуляторы имеют редакторы формул, текстовых комментариев, векторов, матриц и таблиц. Они могут выполнять самые распространенные статистические расчеты, проводить регрессию и вычислять множество статистических параметров для массивов данных. Большую часть таких вычислений они выполняют в «ручном» режиме, но имеют весьма мощные средства программирования и большой объем памяти для хранения множества программ и даже библиотек программ.

Калькуляторы позволяют строить графики различных функций в декартовой и в полярной системе координат, а также графики трехмерных поверхностей и фигур различного типа (рис. 1.4).

Кажется почти невероятным, но они обеспечивают даже анимацию графиков и интерактивное управление углами обзора **3D-фигур** путем их вращения в разные стороны с помощью графического манипулятора. Возможно разбиение экрана на два окна с выводом в каждое окно любой информации. Управление калькуляторами происходит с помощью многочисленных выпадающих меню и встроенного графического манипулятора. Наиболее часто используемые меню вызываются нажатием функциональных клавиш от F1 до F10.

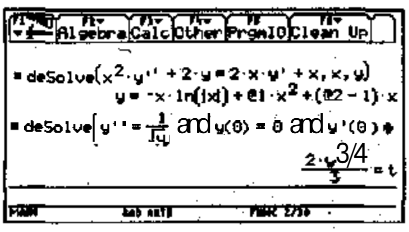


Рис. 1.3. Примеры решения дифференциальных уравнений второго порядка в аналитическом виде (копия экрана)

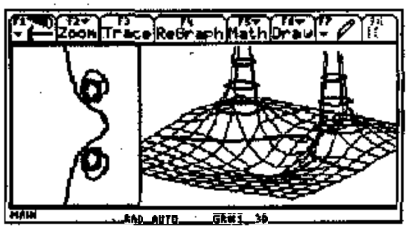


Рис. 1.4. Пример графической визуализации решения в виде контурного и трехмерного графиков (копия экрана)

Широко используются и привычные для пользователей ПК окна (конечно, более скромные, чем в Windows 95/98/NT).

Все это калькуляторы делают с удобствами, которые попросту отсутствуют у персональных компьютеров, даже с установленными на них мощными системами компьютерной математики. Например, для заданной пользователем функции можно мгновенно построить график и провести ее всесторонний анализ. Можно вычислить значение функции в заданной точке, уточнить ее корни, вычислить расстояние между любыми двумя точками графика функции или длину дуги, уточнить значения и места расположения экстремумов и т. д. и т. п. Можно также, задав пределы (как в виде чисел, так и курсором на графике), вычислить определенный интеграл от этой функции и наблюдать его геометрическую интерпретацию — заштрихованную площадь.

Стоят калькуляторы в несколько раз дешевле обычного компьютера (TI-92 Plus, к примеру, стоит около 200 долларов) и имеют перед ним и некоторые эксплуатационные преимущества: меньшие габариты и вес, удобства работы, полная бесшумность, отсутствие вредных излучений, ничтожная потребляемая мощность, длительное время работы (десятки и сотни часов) от одного комплекта батарей. Эти калькуляторы можно взять с собой для работы в самолете и на поезде, на даче или на пляже, не беспокоясь об их электропитании при длительной работе.

Калькуляторы имеют расширенное математическое программное обеспечение, зашитое во флэш-память ПЗУ. Емкость памяти калькуляторов до 890 Кбайт. Они программируются на мощной (свыше 400 инструкций) версии Бейсика с расширенным набором графических функций и команд и функций для аналитических вычислений и преобразований. Любопытно, что Бейсик этих калькуляторов, включая многие функции системы Derive, намного превосходит эту систему в части возможностей программирования управляющих структур и операций ввода-вывода (это уже достижение корпорации Texas Instruments).

Помимо базового программного обеспечения, калькуляторы могут оснащаться пакетами расширения. Например, TI-92 Plus уже поставляется с пакетом расширения по геометрии, позволяющим строить самые различные геометрические объекты и фигуры с интерактивным управлением построениями. Пакет рассчитан на иллюстрацию геометрических построений при изучении курса геометрии в школах и вузах. Ряд пакетов расширения можно скачать с Интернет-сайта фирмы Texas Instruments. Среди них пакеты по статистическим расчетам и даже по машинному расчету электрических и электронных цепей и схем.

Калькуляторы питаются от двух батарей — литиевой батареи для хранения данных и программ и обычных щелочных гальванических элементов для питания процессора, обычной памяти, дисплея и других узлов калькулятора. Смену литиевой батареи надо проводить раз в два года при заправленном отсеке щелочных батарей — это гарантирует сохранность данных при смене батареи. Следует учесть, что для смены щелочных элементов в калькуляторе

TI-92 Plus надо открыть специальный замок и снять нижнюю крышку калькулятора.

В новые калькуляторы заложена возможность полного обновления программного обеспечения, вплоть до замены операционной системы и расширенного математического обеспечения в ПЗУ. Это достигается благодаря применению в ПЗУ электрически перепрограммируемой флэш-памяти. Смена программного обеспечения может выполняться с помощью специального модуля, от другого калькулятора с новым программным обеспечением и через Интернет. На Интернет-сайте (www.ti.com) фирмы Texas Instruments размещена подробная информация о калькуляторах и новейшее программное обеспечение для них.

Калькуляторы имеют миниатюрный штепсельный разъем, обеспечивающий связь с другими калькуляторами или персональными компьютерами по протоколу, принятому для стандартного последовательного интерфейса RS-232. Возможно объединение микрокалькуляторов в локальную сеть.

Для осуществления указанных возможностей нужно приобрести интерфейсный кабель и CD-ROM с программным обеспечением TI GRAPH LINK (стоимостью около 20 долларов). Это программное обеспечение позволяет готовить программы для калькуляторов на ПК, используя наборы операторов и функций калькулятора и шаблоны программ. Такие программы затем можно загружать в калькулятор, отлаживать и использовать. Возможность хранения библиотек программ калькуляторов на компьютере, имеющем дисковую память большого объема, трудно переоценить. Компьютер можно также использовать для распечатки данных, программ и копий экрана калькуляторов.

Сказанное позволяет сделать вывод, что новые калькуляторы не столько являются альтернативой персональным компьютерам в выполнении массовых математических расчетов, сколько их полезным дополнением — своеобразной миниатюрной рабочей станцией для выполнения таких расчетов. Их совместное применение с компьютерами открывает новые возможности в комфортной работе их пользователей в сферах науки, образования и инженерной деятельности. Любопытно, что для этих калькуляторов созданы даже игровые программы.

Автор этой книги, будучи в длительной стажировке (сентябрь — октябрь 2000 года) в США на фирме Wolfram Research Inc., создавшей систему компьютерной математики Mathematica 4, длительное время не мог приобрести массовые микрокалькуляторы TI-89 в магазинах городов Шампейн и Чикаго, ибо эти модели шли буквально нарасхват. В этом нет ничего удивительного — они широко используются в школах, вузах и университетах для большей части расчетов, хотя по уровню оснащения учебных заведений персональными компьютерами последние намного превосходят наши школы и вузы. А калькулятор TI-92 Plus, после упорных месячных поисков, удалось купить случайно, в ничем не примечательном студенческом магазине в городке Иллинойского университета, что расположен в городе Шампейн. Зато пакет для связи этих калькуляторов с ПК можно было купить без проблем.

Возможности систем класса Derive для ПК

И все же, как ни хороши микрокалькуляторы TI-89/92, системы компьютерной математики класса Derive для современных ПК обеспечивают большие функциональные, математические и графические возможности. Система 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 др.;
- 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 т. д.);
- решение задач теории поля и векторного анализа, включая вычисление градиента, дивергенции и потенциала;
- построение двухмерных и трехмерных графиков, графиков функций, заданных в параметрической форме, и графиков в полярной системе координат.

Средства графики Derive позволяют строить двухмерные графики в полярной и декартовой системах координат и трехмерные графики с применением довольно эффективного алгоритма удаления невидимых линий или без него. Графики автоматически масштабируются с целью получения их максимальных (для заданного типа 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 пользоваться возможностями цветной графики. Здесь особо надо отметить новейшую версию Derive 5 под Windows — в ней появилась возможность построения графиков с цветной функциональной окраской, ранее включенная в системы компьютерной математики более высокого уровня.

Таким образом, в начале третьего тысячелетия система Derive стала универсальной математической системой, ориентированной на решение весьма широкого круга математических и научно-технических задач. Многих привлекает то обстоятельство, что Derive и сейчас выпускается в ряде версий, удовлетворяющих весьма разнообразным требованиям пользователей — от упрощенных версий для микрокалькуляторов и компьютеров с операционными системами MS-DOS до версий с вполне современным пользовательским интерфейсом и справочной системой, ориентированных на операционные системы класса Windows 95/98/NT/2000. Таким образом, можно без всякого преувеличения говорить о мире Derive как о совокупности ряда версий этой замечательной системы.

При этом все современные версии Derive являются расширяемыми системами, способными легко адаптироваться под решение специальных задач пользователя. Они поставляются с богатой библиотекой утилит-функций, существенно расширяющей и без того обширные возможности системы. Пользователь может создавать и свои библиотеки, окунувшись в Магию программирования задач компьютерной алгебры. Это характерно даже для версий Derive, используемых в микрокалькуляторах, упомянутых выше.

Правда, расширяемость системы решена 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 достаточен гибкий диск 5,5 дюйма с форматом 1,2 Мбайта (для версии 3.11). Столько же система занимает и на жестком диске.

Может показаться фантастикой, но это факт — маленькая Derive часто справляется с решением аналитических задач малой и средней сложности гораздо быстрее и лучше (с меньшим числом ошибок и отказов от решения), чем ее маститые конкуренты — системы Mathcad под Windows, Mathematica 2 и даже маститая система Maple V под Windows. 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.

Derive 3/4 под MS-DOS 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 принтером. При установке на ПК драйверов кириллицы возможно задание текстовых комментарий на русском языке и даже задание русскоязычных идентификаторов. А системы Derive 4.* и 5.* под Windows могут работать совместно с текстовыми редакторами WordPad и Word 6.0/7.0/8.0, что позволяет готовить прекрасного вида отчеты с русскоязычными текстами и вставками из сессий работы Derive..

Кстати говоря, развитие всех математических систем сейчас идет на пути **ин**-теграции их с современными текстовыми процессорами для создания рабочих мест учащихся, инженеров и ученых.

Для расширения своих возможностей и подготовки библиотек функций Derive имеет входной язык программирования сверхвысокого уровня, манипулирующий с операторами и функциями. Он поразительно прост и не требует от пользователя больших затрат времени на освоение (даже Бейсик куда сложнее, чем этом язык). Описание входного языка Derive дано в последующих уроках данного учебного курса. Там же описаны библиотеки внешних расширений Derive, которые являются 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 лучших реализаций одного из **самых** известных языков высокого уровня — LISP, ориентированных на решение задач искусственного интеллекта и построения «ра**зумных»** экспертных систем [19]. Именно использование языка muLISP придает системе Derive интеллектуальность, которой так не хватает многим малым математическим системам, таким, как Eureka, **Mathcad** (до версии 3.0) или MATLAB. В статье **[23]** описаны особенности реализации системы Derive и языка muLISP.

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

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

Предшественник Derive 3.11 XM — система Derive Professional XM на ПК 486SX (с частотой работы микропроцессора **25** МГц и емкостью оперативного запоминающего устройства (ОЗУ) в 3 Мбайта) «всего» за неделю работы вычислила по этой формуле указанное число точных цифр числа PI! Последние 20 цифр были следующие: 40424876025138l9524. Как говорится, не верите — **проверьте!**

Этот пример наглядно показывает, что Derive может успешно использоваться при решении сложнейших задач теории чисел, при 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.

Derive в образовании и в развитии

Попытки обучить простых смертных аналитической математике в школах и вузах дают, мягко говоря, не слишком впечатляющие результаты. Даже студенты старших курсов вузов, прошедшие обширный и фундаментальный курс высшей математики, проявляют поразительную забывчивость при необходимости применения серьезных математических методов на практике. Речь, конечно, не идет об отдельных вундеркиндах, для которых математика стала целью их жизни и из которых в дальнейшем выходят преподаватели математики вузов и университетов, а об основной массе учащихся. Тех, кто, выучившись в вузах, создают космические корабли и ракеты, разрабатывают и производят новые бытовые устройства и торгуют в лавках пивом и напитками вроде кока-колы.

Увы, но есть множество выпускников технических университетов и вузов, которые за всю свою последующую трудовую жизнь ни разу не вычисляли интеграл или производную какой-либо функции! Не потому ли наша техническая продукция повсеместно (за исключением космонавтики и авиации, где всегда относились к математике с глубочайшим почтением) оказалась самой тяжелой, ненадежной и прожорливой на ресурсы — словом, продукцией, не конкурентоспособной в условиях рынка?

Как ни странно, но в области серьезных математических расчетов до последнего времени компьютеры использовались подобно большим и быстродействующим калькуляторам [1], причем обычным, а не таким «умницам», как TI-89/92/92 Plus. Были созданы и успешно применялись программы-калькуляторы, например уже ставшие хорошо известными простые Eureka и Mercury или калькуляторы, встроенные в операционные системы Windows. Однако они и даже более мощные системы для численных расчетов (например, могучая матричная система MATLAB) были полными «профанами» в символьных и аналитических вычислениях, подвластных математикам — теоретикам и аналитикам, коих издавна считали людьми «не от мира сего». В частности, за их особый стиль мышления и прекрасно развитую математическую интуицию.

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

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

В первой международной конференции по Derive, прошедшей в июле 1994 года в Плимуте (Великобритания) приняло участие более 200 делегатов из 22 стран мира. В июле 2000 года прошла уже четвертая международная конференция «Derive — TI-89/92», закрепившая за Derive статус системы компьютерной алгебры не только для ПК, но и калькуляторов. Открыт Европейский филиал фирмы Soft Warehouse, Inc. — SoftWarehouse Europe (Австрия). В России интересы фирмы Soft Warehouse Inc. представляет МНИИТЦ «Скан».

Серьезным достижением разработчиков Derive в последние годы стал выпуск 32-разрядной профессиональной версии Derive 3.*/4.* XM Professional под MS-DOS, ориентированных на ПК с процессорами 386/486/Pentium. Эти версии работают с расширенной памятью (ОЗУ не менее 2 Мбайт) и практически исключают появление сообщений о нехватке памяти при проведении сложных символьных преобразований, чем иногда грешили обычные версии Derive.

Работа с Derive 3.* и 4.* XM под MS-DOS практически ничем не отличается от работы с ранними версиями Derive. Различие лишь в том, что последние использует ОЗУ емкостью до 640 Кбайт, тогда как Derive XM использует всю память ПК — теоретически до 4 Гбайт (4 биллиона байт). Если прежние версии Derive были ориентированы на решение простых и умеренно сложных задач, то Derive XM может решать самые «крутые» задачи символьной математики с применением обширных библиотек расширений и глубоких рекурсий.

По сравнению с прежними версиями в Derive XM обеспечены 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.

Внедрение математических систем, особенно компьютерной алгебры, в образование выглядит у нас порою просто трагикомично. Педагоги школ и вузов по-разному относятся к автоматизации «святая святых» — математических преобразований. Одни с порога отвергают возможность применения таких систем в образовании, утверждая, что они якобы отучают учащихся от математических навыков и интуиции; другие считают, что нечего учить учащихся большинству, в целом тривиальных, математических преобразований вообще, коль их уже освоили компьютеры.

В действительности неверны 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 смехотворно, как отказ от поездки на поезде на том лишь основании, что «быть не может того, что он едет», или что «пешком ходить полезнее».

Нет никаких серьезных оснований к отказу в использовании возможностей 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 учеников школ и вузов с системой Derive. Их авторы наглядно показывают, что Derive — подлинная находка для педагога-новатора и что эта система эффективно помогает в изучении математики. Первая книга такого рода появилась и у нас [11].

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 грубые ошибки в выполнении символьных вычислений или отказ от их выполнения.

В таких случаях компьютеру нужна квалифицированная помощь — надо подсказать правильное направление преобразований. Но и в этой ситуации система Derive показала себя наилучшим образом — по мизерности таких ошибок эта система обогнала куда более мощные системы такие, как Maple V и Mathematica. Правда, только в случае, когда результаты вычислений представимы через элементарные функции.

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 его применять преподавателем и учащимся. 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 правила «золотой серединки». В этом случае, 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, которые ой как нередко сопровождают 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 отвергаются.

Графика Derive примитивна, если сравнивать ее с графикой больших математических систем, таких, как MATLAB, Mathcad, Maple V или Mathematica 3/4. Тем не менее она вполне достаточна для такой важнейшей задачи, как визуализация вычислений. Между прочим, именно эту проблему президент фирмы Intel Крейг Баретт, ведущей в мире по производству микропроцессоров, назвал визуализацию вычислений одной из важнейших мегатенденций в развитии компьютеров XXI века. Новая версия Derive 5 имеет вполне современные возможности цветной двумерной и трехмерной графики.

Учитывая наличие мощных и интенсивно развиваемых систем символьной математики, таких, как Mathcad 6.0/7.0 PLUS, Mathematica 3/4, Maple V R3/R4, R5, 6 и 7, наиболее предпочтительной сферой применения малых систем Derive являются школы и педагогические университеты и вузы. В технических вузах Derive полезна на младших курсах, хотя ее новейшие версии могут удовлетворить скромные аппетиты студентов и старших курсов.

Отражение мира Derive в Интернете

Начальная Интернет-страница системы Derive

После того как компания Soft Warehouse Inc. влилась в крупную корпорацию Texas Instruments (TI), Интернет-сайт с адресом системы Derive **www.derive.com** автоматически переадресует пользователя Интернетом к странице корпорации TI (www.ti.com), посвященной системе Derive и калькуляторам TI-89/92. Эта страница (на момент подготовки данного раздела книги в январе 2001 года) представлена на рис. 1.5.

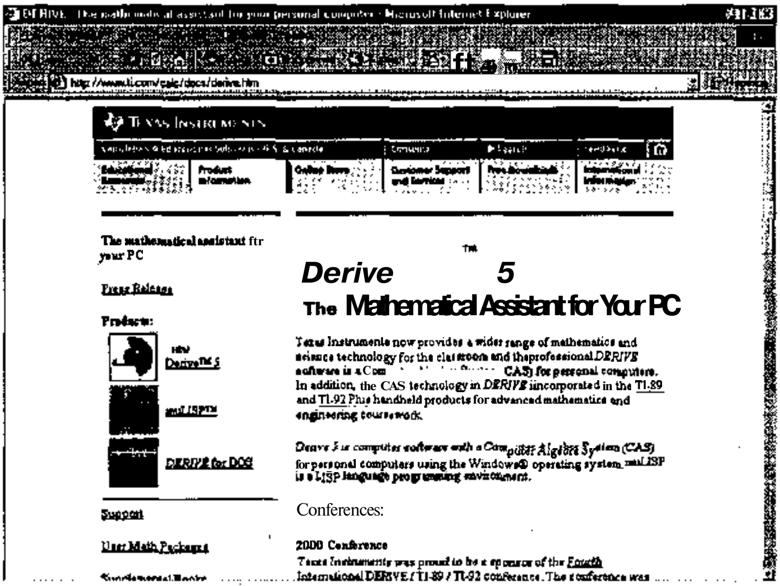


Рис. 1.5. Начальная страница корпорации TI, посвященная системам Derive и калькуляторам TI-89/92

На этой странице рекламируются две версии системы Derive для ПК — версия 4.* под MS-DOS и новейшая версия Derive 5.* под Windows. Здесь же отмечено, что система компьютерной алгебры (CAS) класса Derive используется в новых калькуляторах корпорации TI — TI-89/92. Внизу страницы размещена информация о четвертой международной конференции «Derive — TI-89/92».

В разделе пресс-релизов имеется страница о знаменательном событии — вхождении Soft Warehouse под крылышко корпорации TI (рис. 1.6). Из нее видно, что это событие произошло 6 августа 1999 года.

На Интернет-страницах корпорации TI можно найти данные о примерно 40 издательствах, которые публикуют книги по системам Derive и калькуляторам **TI-89/92/92** Plus.

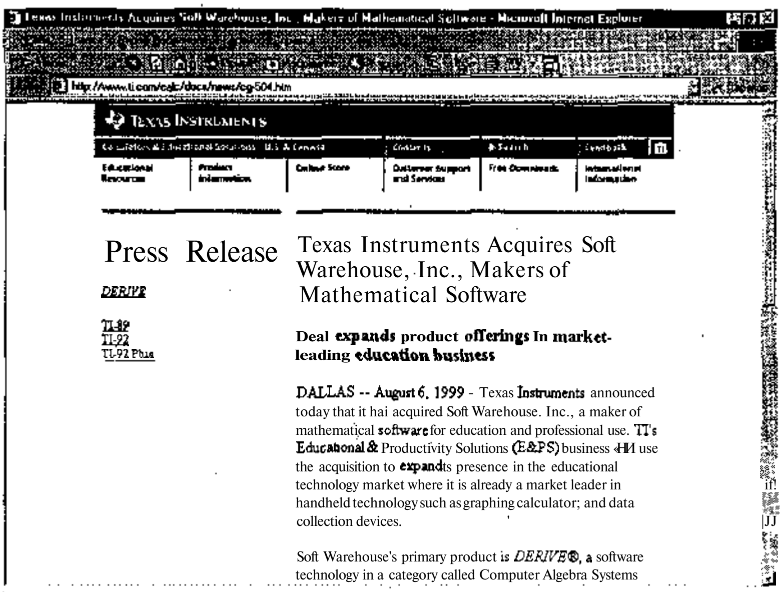


Рис. 1.6. Страница, посвященная слиянию компании Soft Warehouse с корпорацией Texas Instruments

Информация о системах Derive под MS-DOS и Windows

Активизируя гиперссылку Derive ™ 5. можно выйти на страницу с описанием последней версии системы Derive 5 под Windows (рис. 1.7). Эта версия появилась в 2000 году и является наиболее серьезным достижением разра-

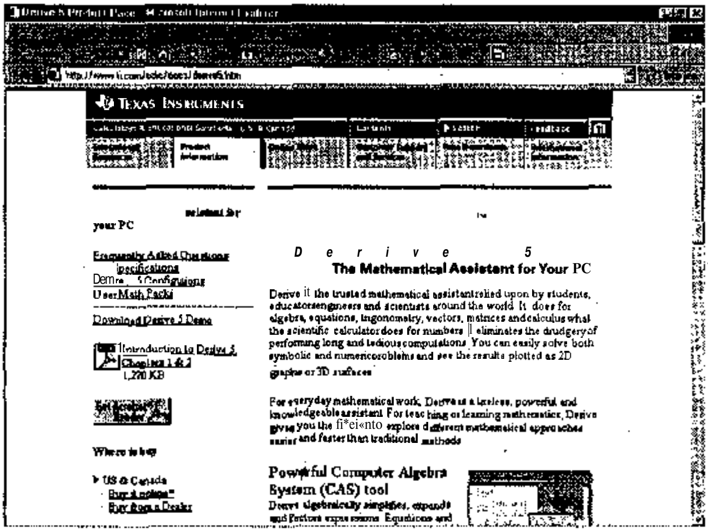


Рис. 1.7. Информация о новой версии системы Derive 5 под Windows ботчиков систем Derive, ориентированных на операционные системы Windows 95/98/NT.

Здесь можно получить полную спецификацию системы, ознакомиться с ответами на наиболее интересные вопросы по ее применению, уточнить аппаратные требования для нормальной эксплуатации системы, загрузить демонстрационную версию системы (со сроком действия в 30 дней), скачать файлы с примерами ее применения и сделать заявку на приобретение системы через Интернет. Аналогичные данные по системе Derive под MS-DOS дает активизация гиперссылки *DERIVE* for DOS.

Информация о конференциях

На Интерненет-сайте, посвященном системе Derive и использующем ее элементы калькуляторам TI-89/92, можно получить информацию о систематически проводимых международных конференциях по этой системе. На рис. 1.8 представлены данные о четвертой конференции «Derive — TI-89/92».

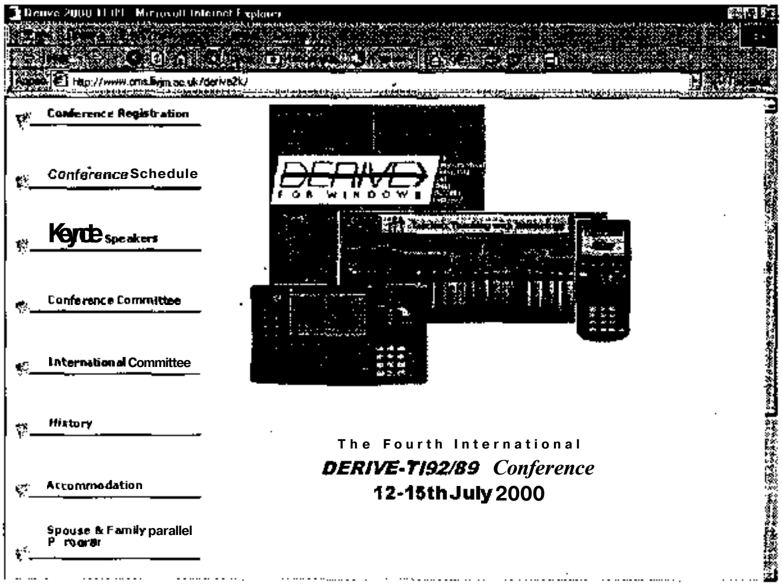


Рис. 1.8. Данные о четвертой международной конференции «Derive — TI-89/92»

Эти крупные конференции сыграли большое значение в популяризации системы Derive и новейших калькуляторов TI-89/92 (особенно в сфере образования). На данной Интернет-странице можно найти данные об оргкомитете конференции, ее программе, условии проживания участников и т. д. Есть и данные о сделанных на ней многочисленных докладах.

Информация о графических калькуляторах TI-89/92/92 Plus

Как видно из представленной выше информации, отныне развитие системы Derive идет в двух важных направлениях — создания систем компьютерной алгебры для ПК и для нового поколения микрокалькуляторов. Активизируя гипертекстовую ссылку TI-89, можно выйти на страницу, посвященную этому типу микрокалькуляторов. Она показана на рис. 1.9.

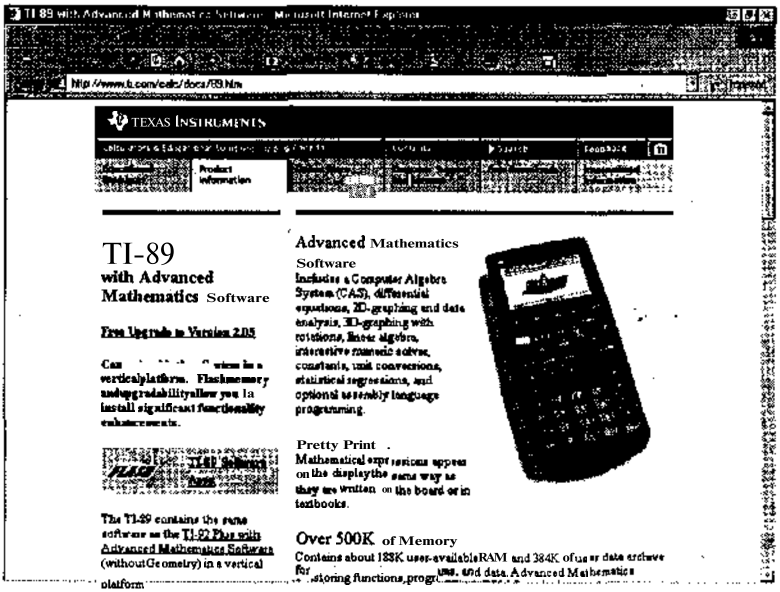


Рис. 1.9. Данные **о** микрокалькуляторах **TI-89**

Аналогично активизируя гипертекстовую ссылку TI-92, можно выйти на Интернет-страницу с описанием (и рекламой) микрокалькуляторов TI-92, которые являются самыми мощными в этом классе устройств (рис. 1.10). Это особенно относится к последним их вариантам TI-92 Plus, оснащенным сменным модулем флэш-памяти, который можно использовать для расширения возможностей этих калькуляторов в той профессиональной сфере, которая нужна их конкретному пользователю.

Обратите внимание на то, что фирма TI выкладывает на сайте обновленную версию как операционной системы этих калькуляторов, так и расширенного математического программного обеспечения на основе системы Derive. Таким образом, через Интернет можно обновлять как базовое, так и прикладное обеспечение микрокалькуляторов TI-89/92/92 Plus. Все это постепенно стирает грани между функциональными возможностями ПК и программируемых микрокалькуляторов.

На одной из Интернет-страниц, посвященных калькуляторам, можно найти статью, в которой приведены данные об объеме выпуска графических калькуляторов для системы образования. Эта страница представлена на

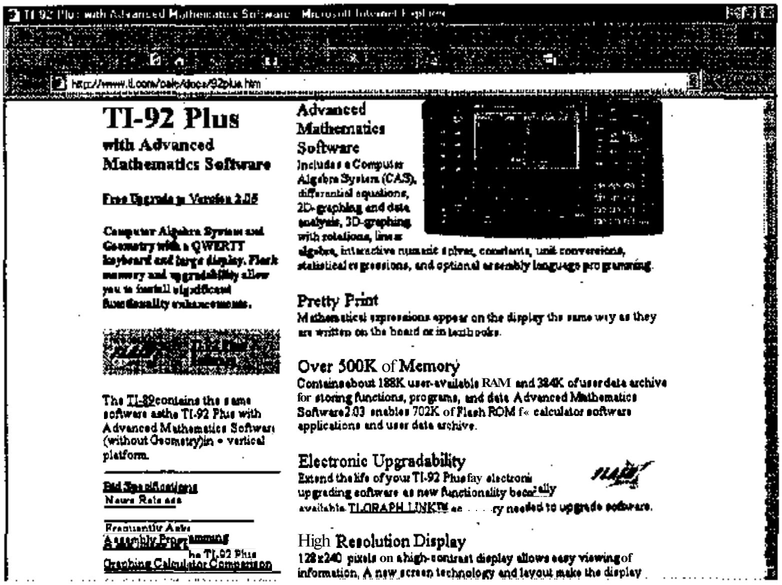


Рис. 1.10. Данные о микрокалькуляторах **TI-92**

рис. 1.11. Из нее следует, что за последние 10 лет корпорация TI выпустила на рынок свыше 20 миллионов графических калькуляторов. Если учесть, что массовым выпуском таких изделий занят ряд других крупных фирм, например Hewlett Packard и **Cassio,** то становится ясным, что этот вид изделий за

Рис. **1.11.** Данные об объеме выпуска графических калькуляторов корпорацией TI

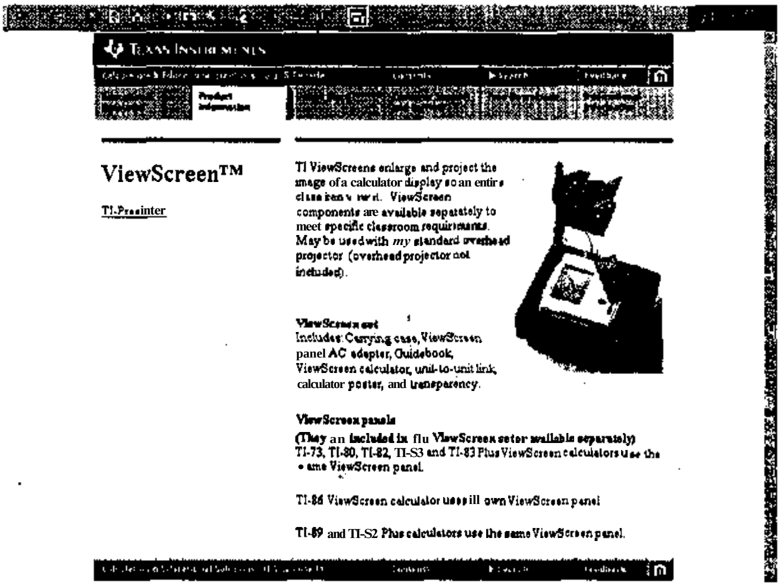


Рис. 1.12. **Интернет-страница** с описанием видеопроектора для просмотра документов микрокалькуляторов на большом экране

рубежом интенсивно развивается и этому ничуть не мешает бум в производстве и применении персональных компьютеров, в том числе и миниатюрных (класса ноутбук). Калькуляторы легко находят свою нишу в средствах информационного обеспечения общества.

Для микрокалькуляторов указанного типа выпускается самое разнообразное периферийное оборудование. Это и адаптер связи калькуляторов с компьютерами, и принтер для печати документов калькуляторов, и разнообразные устройства для подключения калькуляторов к телевизорам и видеопроекторам. Есть и разработки специальных видеопроекторов для графических калькуляторов (рис. **1.12).** Такие проекторы очень удобны при проведении занятий в учебных классах школ и вузов.

Из приведенных данных можно сделать вывод о хорошей поддержке всех версий системы Derive в Интернете. Это, несомненно, способствует их популяризации и широкому применению во всем мире.

С *чем мы познакомились:*

- С историей появления и ролью систем Derive.
- С созданием калькуляторов **TI-89/92/92** Plus фирмы Texas Instruments.
- С возможностями систем класса Derive для ПК.
- С применением Derive в образовании и в развитии.
- С отражением мира Derive в Интернете.

Урок 2. Работа с Derive под MS-DOS

- Подготовка к работе
- Меню систем Derive под MS-DOS
- Редактирование и ввод выражений (Author)
- Примеры работы с системой Derive XM
- Построение математических выражений (Build)
- Уничтожение строк (Remove)
- Автоматическая очистка оперативной памяти
- Задание глобальных опций (Options)
- Работа с внешними устройствами (Transfer)
- Перемещение строк (moVe)
- Работа с окнами (Window)
- Работа с помощью (Help)
- Перемещение по строкам документа (Jump)
- Подсказки и индикация ошибок
- Выход из системы (Quit)

Подготовка к работе

Установка и загрузка системы

Система Derive под MS-DOS в настоящее время представлена версиями 3 и 4 с их модификациями. Различия между ними настолько незначительны, что мы будем рассматривать их как одну версию и будем называть ее Derive 3.*/4.* XM или просто Derive 3/4. Обозначение XM (от слов extended Memory — расширенная память) введено для профессиональных версий Derive, поддерживающих работу с ОЗУ большого объема — до 4 Гбайт, что существенно расширяет степень сложности решаемых задач.

Системы Derive под MS-DOS (без приставки XM) могут работать на ПК с объемом ОЗУ от 512 Кбайт до 640 Кбайт, оснащенных одним дисководом с двухсторонним считыванием. Derive XM под MS-DOS требует минимального объема оперативной памяти — 2 Мбайта — и в принципе 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).

Инсталляция систем происходит самым обычным образом. Версии Derive под MS-DOS на наших CD-ROM обычно вообще не требуют инсталляции. Достаточно просто скопировать директорию с Derive на жесткий диск ПК. Для этой же цели обычно используются и возможности администратора жесткого диска Norton Commander или другой, ему подобной программы. Пуск системы при этом выполняется просто указанием на исполняемый файл derive.exe.

Проблемы с запуском в среде Windows

Derive 3/4 XM под MS-DOS лучше использовать по своему прямому назначению — в среде MS-DOS. Однако если пользователь пользуется системой редко и жесткий диск его компьютера с Windows забит до предела, то можно запускать Derive 3/4 и из Windows. При этом возможны некоторые проблемы.

Одна из проблем заключается в ограниченных возможностях управления окном. Например, его невозможно плавно растягивать. Окно выводится нередко искаженным — удлиненным по горизонтали. Связано это с тем, что сразу после загрузки Derive 3/4 работает в текстовом формате. При этом графики строятся очень грубо — по точкам в текстовом знакоместе, а вывод математических выражений весьма отдаленно напоминает стандартный (с математической нотацией). От этих недостатков можно легко избавиться, переключив систему в режим графического вывода с помощью опций дисплея. Как это сделать, описано в следующем уроке.

Еще один недостаток — блокирование работы переключателя языков (с русского на английский и наоборот), который расположен в строке запуска приложений Windows 95/98/NT. При этом англоязычная система Derive упорно не желает переключаться на английский язык ввода символов. От этого недостатка также нетрудно избавиться — загрузите (с помощью файлов autoexec.bat или config.sys) подходящий драйвер клавиатуры и пользуйтесь тем набором клавиш переключения языков, который задается этим драйвером.

Более серьезный и, к сожалению, неустранимый недостаток — искажение некоторых символов (например, мнимой единицы и основания натурального логарифма). Оно связано с тем, что эти символы берутся из дополнительной кодовой таблицы (коды от 128 до 255), которая при русификации заменяется таблицей символов кириллицы. С этим недостатком приходится мириться, если вы хотите иметь возможность задания программных комментарий на русском языке.

Достоинства и недостатки Derive под MS-DOS

Можно ли возможность работы без жесткого диска рассматривать как главное достоинство Derive под MS-DOS? Разумеется, нет. Вряд ли сейчас можно встретить ПК без жесткого диска. И тем не менее идут упорные разговоры о создании таких дешевых ПК, ориентированных в основном на работу в Интернете и не имеющих дорогих жестких дисков. Так что кто знает, может, владельцы этих «почтовых ящиков» с благодарностью вспомнят, что есть довольно мощная математическая система Derive, вполне размещаемая на обычной дискете.

Предшествующие версии Derive под MS-DOS (от 2.0 до 2.6) вполне довольствовались ОЗУ емкостью даже менее 640 Кбайт. Кстати, весь материал этой книги (кроме описания интерфейса систем под Windows) вполне подойдет для знакомства с ними. Однако версия 3.11 XM требует уже памяти ОЗУ не менее 2 Мбайт. К тому же это уже 32-разрядная версия. Так что для работы с ней потребуются ПК с микропроцессором не хуже 386. Впрочем, есть возможность запускать эту версию и в 16-разрядном режиме работы.

Можно считать, что основными достоинствами версий Derive под MS-DOS являются следующие их качества:

- минимально возможные требования к аппаратным ресурсам ПК;
- возможность работы на ПК с 16-разрядными процессорами;
- предельно простой и легкий в изучении интерфейс пользователя;
- повышение сложности решаемых задач при заданном объеме памяти.

Из недостатков Derive 3/4 под MS-DOS (в сравнении с версиями под Windows) можно отметить следующие недостатки:

- проблемы при запуске из Windows (см. выше);
- ущербный в своей простоте интерфейс пользователя;
- устаревший и архаичный стиль работы;
- невозможность управления от графического манипулятора — мыши;
- невозможность создания документов в форме полноценных ноутбуков (записных книжек);
- ограниченные возможности графики, особенно трехмерной;
- проблемы с печатью документов и графики;
- слабая справочная база данных.

Для пользователей современными ПК с объемом ОЗУ 8 Мбайт и выше применение Derive 3/4 под MS-DOS кажется архаичным. Но упрямая статистика показывает, что даже в богатых странах парк ПК на стареньких процессорах 8086, 286, 386, 486 и «старых» Pentium во много раз превосходит по своему размеру парк современных ПК на базе микропроцессоров Celeron, Pentium II, Pentium III, Pentium 4 и их подобных. А потому даже систему Derive 3.* под MS-DOS рано списывать со счетов и неслучайно разработчики Derive продолжают выпускать улучшенные и новые версии 4.* этой системы под MS-DOS.

Расширяемость систем **Derive под MS-DOS**

Как отмечалось, Derive является расширяемой системой. Ее расширения можно загрузить сразу 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 текстов их документов на экран дисплея. При этом загрузка идет намного быстрее, чем в случае загрузки с выводом сообщений на экран дисплея.

Поскольку файлы документов и библиотечные файлы Derive 3/4 имеют текстовый формат, то заинтересованный пользователь может легко составить из имеющихся файлов свою библиотеку функций и дополнить ее определениями не найденных функций. Не стоит включать в такую библиотеку слишком много функций — их должно быть не более чем несколько десятков, хотя сама система воспринимает куда большее число определений новых функций и ограничений на их число в новых версиях практически сняты.

Использование демонстрационных файлов

Система содержит также специальные демонстрационные файлы с именами типа *.dmo. Их также можно загрузить, используя команду **Load** с опцией **Demo.** Демонстрационные файлы можно исполнять, используя команду **Simplify.** В меню USER можно найти ряд расширений Derive и демонстрационных файлов, созданных пользователями — наиболее ярыми поклонниками этой системы. Вы можете войти в их число, направив свои разработки в адрес фирмы Soft Warehouse (ее адреса и телефоны можно найти в справочной системе помощи Derive). Кроме того, в этой директории имеются текстовые файлы дополнительной документации.

Перед работой с Derive рекомендуется просмотреть два текстовых файла: derive.doc (краткая документация по системе есть не во всех поставках) 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.

Меню систем Derive под MS-DOS

Обзор позиций меню

После загрузки систем Derive под MS-DOS на экране дисплея появляется окно системы (рис. 2.1). Оно содержит название системы, номер ее версии, данные о фирме-разработчике и предупреждение о нежелательности использования нелегальных копий Derive.

Далее пользователю напоминается, что нажатие клавиши H (от Help — помощь) позволяет ввести режим помощи. Это рекомендуется сделать уже при первом знакомстве с системой.

Двойная черта делит экран на два окна. Верхнее (большое) окно используется для ввода математических выражений и комментарий, а также для вывода результатов вычислений в символьном или числовом виде. Будем называть его окном редактирования документов или просто окном докумен-

```
                         D E R I V E  X M
                     ft Mathematical Assistant

                           Version 3.11


                 Copyright (C) i988 through 1995 by
                        Soft Warehouse, Inc.
                  3660 Waialae Avenue, Suite 304
                 Honolulu, Hawaii, 96816-3236, USA

Please do not nake illegal copies of DERIVE! This software is not shareware or
freeware.  It is not to be published on bulletin boards or distributed by any
other means without written permission fron Soft Uarehouse, Inc.

For technical support or if you know of any person or company distributing
DERIVE as shareware or freeware, please write us at the above address or send a
fax to (808) 735-1105.

                          Press H for help



COMMAND: Author Build Calculus Declare Expand Factor Help Junp soLve Manage
         Options Plot Quit Remove Simplify Transfer Unremove moVe Window approX
Enter option                                                         Derive XM
                                Free:100% Ins                          Algebra
```

Рис. 2.1. Экран системы Derive 3.11 XM

тов, как это стало общепринятым при описании систем компьютерной математики.

Нижнее окно (окно меню) имеет всего четыре строки. Две из них содержат перечень позиций (команд) основного меню системы. Третья строка содержит краткую подсказку. 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), включение или отсутствие режима вставки (Insert) и о текущем режиме работы системы.

Меню Derive XM имеет явно необычное для современных программ расположение, характерное для устаревших программных продуктов. Оно находится внизу экрана под окном документов. Меню имеет 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 системы помощи с кратким описанием возможностей Derive;

Jump — переход (прыжок) к заданной строке;

soLve — решение математических уравнений;

Manage — управление вычислительным процессом;

Options — задание различных опций (установок) системы;

Plot — управление графическими средствами системы;

Quit — выход из системы (конец 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, **Calculus)** появляется дополнительное подменю с рядом команд.

Пользователи современными 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 работа с этим «зверьком» стала для вас жизненной необходимостью, то переходите к работе с Derive 4/5 под Windows. А пока настройтесь на изрядно позабытый стиль — управление только с помощью клавиатуры.

Переключение позиций меню

Выделение одной из позиций главного меню может производиться следующими клавишами:

Перемещение вправо	**tab, Ctrl-I**
Перемещение влевр	**Backspace, Ctrl-H, Ctrl-B**
Перемещение в позицию	**Author Ctrl-[**

Здесь знак **←-→** означает совместное нажатие двух относящихся к нему клавиш. Сам по себе этот значок не вводится.

Альтернативный и более быстрый (типично **калькуляторный)** способ выбора нужной позиции меню заключается в нажатии выделенной (большой) буквы имени позиции. В большинстве случаев (но не **всегда)** это первая буква имени. При этом соответствующая команда выполняется сразу и не требуется нажатия клавиши **Enter.** Этот способ является наиболее предпочтительным.

Действие 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. При этом клавиша **Ins** или комбинация клавиш **Ctrl+V** задает включение и выключение режима вставки. Команда **approX** (аппроксимация) позволяет вычислить это выражение, а команда **Simplify** (упрощение) — упростить выражение.

Другие команды порождают появление дополнительных подменю. Например, команда **Transfer** порождает 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 порождать подменю.

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

О русификации систем Derive

У частично русифицированной версии Derive 4.* под MS-DOS (см. рис. 2.2) система помощи (хелпинг) выполнена на русском языке, что заметно облегчает работу с системой для наших пользователей. Оригинальная версия Derive имеет, естественно, англоязычную справочную систему. Поскольку команды

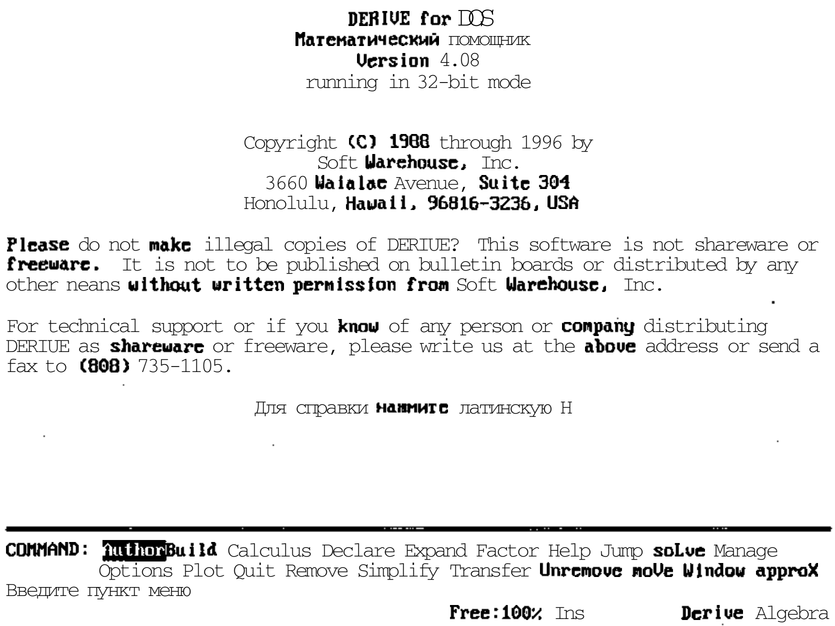


Рис. 2.2. Окно частично русифицированной системы Derive 4.08 **XM**

главного меню одинаковы как для русифицированных, так и англоязычных версий, в дальнейшем мы не будем обращать внимания на тонкости русификации, если, конечно, они не имеют принципиального значения.

Хорошо, что русификаторы системы Derive обеспечили перевод на русский язык только справки и некоторых типовых команд интерфейса пользователя. Все команды меню и символьной математики остались англоязычными, что исключило путаницу в их трактовке и засорение русского языка неточными терминами. Надо учитывать, что язык команд символьной математики, как и математики вообще, уже давно стал интернациональным, а точнее говоря, английским, поскольку первые системы компьютерной математики для ПК появились на Западе и все мы привыкли к англоязычному написанию их команд и функций.

Редактирование и ввод выражений (Author)

Ввод выражений

Но вернемся к описанию работы с системой Derive под MS-DOS. Прежде чем начинать какие-либо вычисления, необходимо задать (ввести) 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 иные встроенные функции Derive. В последнем случае принципиально необходимы средства редактирования выражений перед их вычислением.

Математические выражения составляются 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 — 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 — могут быть функции и ряда 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, большими буквами.

По умолчанию используются однобуквенные переменные, например x, y и z и, кроме того, знак умножения заменяется пробелом. Так что xyz 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 xyz или X_coordinate. Надо учитывать, что знаки кириллицы (русского языка) для записи имен переменных (их идентификаторов) не допустимы.

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

Кроме того, при записи математических выражений важное значение имеет приоритет операций. Он выше у сложных функций и низший 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 **Author** и соответствующей позиции главного меню. После ее ввода появляется запрос

AUTHOR expression: <редактируемое выражение>

После него нужно ввести какое-либо выражение, отредактировать его и зафиксировать ввод нажатием клавиши **Enter.** Здесь и далее перевод сообщений на английском языке дается в скобках (он, естественно, на экран дисплея не выводится). Для ввода в область редактирования выделенных фрагментов выражений (или целиком выражений) можно использовать клавиши F3 и F4 (более полное описание редактирования с применением клавиш клавиатуры описано ниже).

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 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).

Клавиши редактирования и управления курсором

Редактор математических выражений в Derive самый примитивный и простой — строчный. Это значит, что любое, даже самое сложное выражение придется вводить в одной строке. Это предполагает знание ввода выражений в таком виде. Надо полагать, что пользователи математическими системами уже знают, как осуществляется такой ввод: к примеру, он характерен для всех языков программирования, например Бейсике или Паскале. Важно отметить, что запись выражений в текстовом формате в файлах Derive с расширением .mth происходит именно в той форме, в которой выражения вводятся.

Главное, что надо знать при редактировании в строчном редакторе, — это то, что вводимые символы располагаются в месте расположения текстового маркера (курсора), имеющего вид вертикальной черты. Клавиша вставки Ins позволяет устанавливать два режима работы — вставки нового 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 **Author** 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 — вставить выделенное выражение в строку редактирования;

F4 — вставить выделенное выражение в строку редактирования, 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 (знак = означает наличие альтернативной клавиши или комбинации клавиш):

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.

Разные 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.

Выделение выражений и подвыражений

Любое выражение или часть его можно выделить. Выделенное выражение отображается инверсными по цвету знаками с инверсным фоном. Например, если нормально ваши знаки представляются белыми на черном фоне, то выделенное выражение будет представлено черными знаками на белом фоне. Для управления областью выделения используются представленные ниже комбинации клавиш.

Команды выделения выражений:
Ctrl-E или T — перейти вверх на одно выражение;
Ctrl-X или ↓ — 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 или **Ctrl-Home** — перейти вверх на первое выражение;
Ctrl-PgDn или **Ctrl-End** — перейти вниз на последнее выражение;
Ctrl-F или **Ctrl→** — передвинуть выражение влево на пол-экрана;
Ctrl-A или **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 t — перейти на один уровень вверх;
Ctrl-X или ↓ — перейти на один уровень вниз.

Выделение выражений или подвыражений — важный момент работы с системой Derive. Поэтому настоятельно рекомендуется поупражняться в выделении частей какого-нибудь достаточно сложного выражения и освоить технику выделений как его целиком, так и отдельных его частей.

Основы ввода и редактирования выражений

На рис. 2.3 представлен экран Derive после ввода двух математических выражений и в процессе набора третьего выражения. Последнее из введенных выражений выделяется — оно размещается в четырехугольнике и представлено в инверсном цвете. Как только отмечалось, область выделения можно менять. Ввод в командной строке возможен после исполнения команды **Author** и выглядит как последовательный набор с клавиатуры следующих выражений:

```
          2
#1:   a·x  + b·x + c
      ──────────────
       2·a·b·c·x

            SIN(x)
      ──────────────────
#2:          2        2
      x·(SIN(x)  + COS(x) )

AUTHOR expression: SIN(x)/x_
Enter expression (press F1 for help)                Derive XM
User                        Free:100%  InsLin         Algebra
```

Рис. 2.3. Ввод двух математических выражений

(a*x^2+b*x+c)/(2*a*b*c*x)

и

SIN(x)/(x*(SIN(x)^2+COS(x)^2))

Ввод каждого выражения завершается нажатием клавиши **Enter,** после чего выражения в преобразованном виде появляются в окне документа. С выделенным выражением можно выполнять различные операции. Например, на рис. 2.4 показан процесс подготовки к упрощению выражения в строке #2 (знак # используется для указания нужной строки) документа. Для выделения этой строки надо воспользоваться клавишей **перемещения курсора вверх** Т. После этого, используя, например, клавишу **Tab,** нужно выделить команду упрощения **Simplify** в меню. Этот момент представлен на рис.2.4.

```
          2
      a·x  + b·x + c
#1:   ──────────────
       2·a·b·c·x

            SIN(x)
      ──────────────────
#2:          2        2
      x·(SIN(x)  + COS(x) )

       SIN(x)
#3:   ───────
         x

COMMAND: Author Build Calculus Declare Expand Factor Help Junp soLve Manage
         Options Plot Quit Renove Simplify Transfer Unremove moVe Window approX
Enter option                                                   Derive XM
User                           Free:100% Ins                    Algebra
```

Рис. 2.4. Подготовка к исполнению команды Simplify для выражения во второй строке

Теперь надо нажать клавишу **Enter.** В окне меню появится запрос на номер строки, для которой будет исполнена команда **Simplify.** Поскольку выражение в строке #2 уже было выделено, то соответствующий номер строки уже будет указан после этого запроса (рис. 2.5).

```
          2
#1:   a·x  • b·x + c
     ─────────────────
       2·a·b·c·x

             SIN(x)
     ──────────────────────
#2:            2         2
     x·(SIN(x)  • COS(x) )

       SIN(x)
#3:  ─────────
         x

SIMPLIFY expression: #2

Enter expression                                         Derive XM
User                         Free:100%   Lin             Algebra
```

Рис. 2.5. Запрос на выполнение операции Simplify для выражения во второй строке

Так что остается нажать клавишу **Enter,** и результат выполнения операции **Simplify** над выражением #2 появится в новой строке документа — #4. Это показано на рис. 2.5.

Обратите внимание на то, что выражения в строках #3 и #4 абсолютно идентичны. Но выражение в строке #3 результат прямого ввода, тогда как выражение в строке #4 результат упрощения выражения в строке #2. Надо полагать, что читатель понял, что сделала система Derive в нашем первом эксперименте, — она учла, что сумма квадратов синуса и косинуса при любом аргументе x равна 1 и упростила выражение #2 с учетом этого положения.

Однако совершенно аналогичный вид выражений в строках #3 и #4 может спутать пользователя, поскольку не всегда ясно, каким образом получено то или иное выражение. Это обстоятельство является одним из недостатков версий Derive под MS-DOS. Для его устранения можно вводить текстовые

```
          2
#1:   a·x  + b·x * c
     ─────────────────
       2·a·b·c·x

             SIN(x)
     ──────────────────────
#2:            2         2
     x·(SIN(x)  • COS(x) )

       SIN(x)
#3:  ─────────
         x

       SIN(x)
#4:  ─────────
         x

COMMAND: Author Build Calculus Declare Expand Factor Help Jump soLve Manage
         Options Plot Quit Remove Simplify Transfer Unremove moVe Window approX
Compute time: 6.8 seconds                                Derive XM
Simp(#2)                      Free:100% Ins              Algebra
```

Рис. 2.6. Выполнение **операции** Simplify над выражением #2

#1: $\dfrac{a\cdot x^2 + b\cdot x + c}{2\cdot a\cdot b\cdot c\cdot x}$

#2: $\dfrac{SIN(x)}{x\cdot (SIN(x)^2 + COS(x)^2)}$

#3: $\dfrac{SIN(x)}{x}$

#4: $\dfrac{SIN(x)}{x}$

COMMAND: Author Build Calculus Declare Expand Factor Help Jump solve Manage
Options Plot Quit Remove Simplify Transfer Unremove moVe Window approX
Compute time: 0.0 seconds Derive XM
User Free:100% Ins Algebra

Рис. 2.7. Пример выделения числителя в выражении в строке #1

комментарии, что будет рассмотрено несколько позднее. Такие комментарии вводятся в кавычках и не исполняются.

Обычно при работе с Derive бывает выделена последняя строка в окне выражений. Однако, как отмечалось, с помощью указанных выше клавиш можно перемещать выделение от одной строки к другой и даже от одной части выражения на другую. Например, чтобы выделить числитель выражения #1, достаточно нажимать клавишу перемещения курсора вверх до выделения всего выражения, а затем нажать клавишу ← для выделения только числителя. Результат этого показан на рис. 2.7.

С выделенным выражением можно выполнять различные операции. На рис. 2.7 представлен случай, когда задается операция для выделенной части выражения — используется активизация позиции **Calculus** меню.

#1: $\dfrac{a\cdot x^2 + b\cdot x + c}{2\cdot a\cdot b\cdot c\cdot x}$

#2: $\dfrac{SIN(x)}{x\cdot (SIN(x)^2 + COS(x)^2)}$

#3: $\dfrac{SIN(x)}{x}$

#4: $\dfrac{SIN(x)}{x}$

CALCULUS: Differentiate Integrate Limit Product Sum Taylor Vector

Compute time: 0.0 seconds Derive XM
User Free:100% Ins Algebra

Рис. 2.8. Выбор операции дифференцирования

```
               2
          a·x  + b·x + c
#1:      ────────────────
           2·a·b·c·x

                 SIN(x)
#2:      ─────────────────────────
                  2         2
         x·(SIN(x)  + COS(x) )

         SIN(x)
#3:      ──────
           x

         SIN(x)
#4:      ──────
           x

CALCULUS DIFFERENTIATE expression: #1

Enter expression                                              Derive XM
User                               Free:100%      Lin         Algebra
```

Рис. 2.9. Запрос на указание номера строки дифференцируемого выражения

Если теперь нажать клавишу **Enter,** то в меню появится список команд, доступных из позиции меню **Calculus.** Этот момент показан на рис. 2.8. Видно, что в данном случае выделена операция дифференцирования **Differentiate.**

Если нажать клавишу **Enter,** то появится запрос на то, к какой строке будет применена операция выражения (рис. 2.9). Заметим, что операция будет применена только в отношении выделенной части выражения в соответствующей строки.

Нажав клавишу **Enter,** мы увидим в окне-меню запрос на то, по какой переменной должна выполняться операция дифференцирования (рис. 2.10). При этом система задает по умолчанию эту операцию по пере-

```
               2
          a·x  + b·x + c
#1:      ────────────────
           2·a·b·c·x

                 SIN(x)
#2:      ─────────────────────────
                  2         2
         x·(SIN(x)  + COS(x) )

         SIN(x)
#3:      ──────
           x

         SIN(x)
#4:      ──────
           x

CALCULUS DIFFERENTIATE variable: x

Enter variable                                                Derive XM
User                                 Free:100% Ins            Algebra
```

Рис. 2.10. Запрос на то, по какой переменной будет выполнена операция дифференцирования

```
          2
      a·x  + b·x + c
#1:  ──────────────────
         2·a·b·c·x

            SIN(x)
     ─────────────────────
#2:            2        2
     x·(SIN(x)  + COS(x) )

      SIN(x)
#3:  ────────
        x

      SIN(x)
#4:  ────────
        x

CALCULUS DIFFERENTIATE: Order: 1

Enter expression                                       Derive XП
User                            Free:100% Ins          Algebra
```

Рис. 2.11. Запрос на ввод порядка вычисляемой производной

менной x. В принципе можно указать и другую переменную, например, a, b или c.

Выберем все же переменную x и нажмем клавишу ввода Enter. На этот раз снова поступит запрос о том, какого порядка производную вы собираетесь найти (рис. 2.11). При этом по умолчанию система задает вычисление первой производной. Разумеется, если это вам надо, вы можете указать иной порядок вычисления производной.

Нажав клавишу Enter, можно увидеть, что в очередной строке #5 появилась запись операции дифференцирования (рис. 2.12). Заметим, что это еще не результат дифференцирования.

```
          2
      a·x  + b·x + c
#1:  ──────────────────
         2·a·b·c·x

            SIN(x)
     ─────────────────────
#2:            2        2
     x·(SIN(x)  + COS(x) )

      SIN(x)
#3:  ────────
        x

      SIN(x)
#4:  ────────
        x

     d      2
#5:  ── (a·x  + b·x + c)
     dx

COMMAND: Author Build Calculus Declare Expand Factor Help Jump soLve Manage
         Options Plot Quit Remove Simplify Transfer Unremove moVe Window approX
Enter option                                           Derive XИ
Dif(#1',x)                      Free:100% Ins          Algebra
```

Рис. 2.12. Подготовка к последней операции дифференцирования

```
          2
     a·x   · b-x * c
#1:  ─────────────────
       2·a·b·c·x

              SIN(x)
     ─────────────────────
»2:              z        z
     x·(SIN(x)  * COS(x) )

     SIN(x)
ИЗ:  ──────
       X

     SIN(x)
#4:  ──────
       X

     d      2
#5:  — (a-x  + b·x + c)
     dx

#6:  2·a·x + b
COMMAND: Author Build Calculus Declare Expand Factor Help Jump soLve Manage
         Options Plot Quit Remove Simplify Transfer Unremove moVe Window approX
Conpute tine: 0.0 seconds                                          Derive XM
Simp(#5)                              Free:100% Ins                  Algebra
```

Рис. 2.13. Вычисление производной **по** выражению в строке #5

Если теперь исполнить команду **Simplify** (как это делается, описано выше), будет получена производная знаменателя выражения #1, представленная к вычислению выражением #4 (**рис.** 2.13).

Таким образом, выполняются типовые вычисления в системе Derive. В дальнейшем будет приведено множество других примеров проведения вычислений. При этом даже пользователям Derive под MS-DOS полезно просмотреть и выполнить примеры, приведенные в уроках, в которых описана работа в версиях системы Derive под Windows.

Ввод греческих и русских букв в 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)	тета (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 видеоадаптера 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. В связи с этим знаки кириллицы следует применять только для записи комментариев на русском языке. Вообще говоря, если для вас главное — корректная запись математических выражений, то следует отказаться от русификации Derive и загрузки драйвера русскоязычных шрифтов, перейдя к применению оригинальной англоязычной версии Derive.

Понятие о **выражениях,** операторах и **функциях**

Центральным понятием систем компьютерной математики является выражение. Выражение можно рассматривать как запись некоторого вычислительного процесса на том или ином языке. 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, например sin(l), возвращает значение преобразованного параметра (в данном примере число 1 — фактический параметр функции, а sin(l) возвращает синус 1 рад.). Поэтому функции можно использовать в составе математических выражений, например: exp(2*sin(l)).

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 обзора виде.

Derive использует обычный приоритет операций — первыми выполняются более сложные операции. Например, при вычислении арифметических выражений вначале вычисляются значения функций, затем операции возведения в степень, операции умножения и деления и, наконец, операции сложения и вычитания. Поэтому, к примеру, Derive вычисляет 2+3*4 как 14, а не как (2+3)*4=20. В спорных случаях, например **2^3/4** (вроде 2 в степени 3/4), надо вводить скобки — **2^(3/4).**

Примеры работы с системой Derive XM

При работе с математическими пакетами всегда встречается противоречие между естественной (или точнее, общепринятой) формой записью выражения и набором символов, которые может вводить клавиатура ПК и отображать его дисплей. Например, на клавиатуре нет знаков квадратного корня, производной, интеграла и т. д., хотя знакогенератор дисплея частично эти знаки содержит. Поэтому пользователи ПК привыкли к различным обозначениям: например, exp(x) для числа е в степени x, sqr(x) или 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 [5]) воспринимается уже 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 (например Бейсике). 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:

```
sqrt((2.3 — 8.5)^2 + (4 — 0.7)^2)
```

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

```
                     2             2
1:        √ ((2.3 -8.5)   +{4-0.7)   )
```

Обратите внимание, что слово **sqrt** исчезло и вместо него на экране дисплея появился **знак квадратного корня** V. При этом для распространения его действия на все выражение используются общие круглые скобки.

Исчез и знак возведения в степень **^** — эта операция также представлена в общепринятом виде. Впрочем, повод для замечаний есть — например, знак квадратного корня не охватывает все выражение (ничего страшного в этом 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 **approX** (для этого достаточно нажать клавишу X). В ответ появится запрос

APPROXexpression: #1 <Вычисляетсявыражение: #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 арифметическое выражение

2+(3+4)/(5–1)

На экране дисплея появится

```
          3 + 4
3:     2+ ______
          5 - 1

4:     3.75
```

Обратите внимание, что знак деления в виде наклонной черты / оказался замененным на горизонтальную черту (в текстовом формате вывода она состоит из черточек, а в графическом формате будет выглядеть как непрерывная черта). Полезно обратить внимание, что выражение в строке #3 построено с учетом общепринятого приоритета математических операций с учетом действия изменяющих приоритет скобок.

Последний пример демонстрирует вычисление факториала:

```
5:     10!
                  6
6:     3.6288 10
```

Результат при этом оказался представленным не совсем в 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 **ApproX** 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 показаны примеры на такие вычисления с введенными текстовыми комментариями:

```
1: "Зададим выражение (команда Author):"
                2          2
2:     SIN(x)   + COS (x)

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

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 **approX** 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, показанный в строке **8,** может показаться ошеломляющим — оказывается, простое выражение #6 раскладывается на целых пять **членов!** И таких примеров, в том числе куда более сложных, 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, в этом примере мы рассмотрели три формы одного и того же алгебраического выражения. В действительности этих форм может быть намного больше. Не следует полагать, что любое выражение из математического справочника Derive тут же сведет к аналогичному тому, который дан в справочниках. 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 библиотеке Derive 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 3/4 и ориентирован главным образом на преобразования выражений с элементарными функциями. Именно поэтому пользователи-математики высшей квалификации отдают предпочтение этим пакетам, несмотря на их чрезмерные запросы по ресурсам ПК, медлительность в работе и даже возможные ошибки, — как говорится в одной из наших поговорок, «не ошибается только тот, кто ничего не делает».

Текст, представленный в примерах на символьные операции (и в большинстве последующих примеров), 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 данной книге называется документом, а не программой. Как уже говорилось, он отражает рабочую сессию вычислений.

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. По этой причине систему Derive относят к системам без программирования. Далее будет показано, что это не совсем правомерно и при необходимости пользователь может запрограммировать решение своих специфических задач на входном языке системы сверхвысокого уровня.

В дальнейшем при описании вычислений в среде Derive под MS-DOS мы будем приводить большинство примеров в виде, характерном для рассмотренных выше примеров, т. е. 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 **Author.**

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 подключения 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 исполнении 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 п 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 создано заданное выражение. По окончании конструирования выражения необходимо в перечне операторов и функций указать команду **Done.** В ответ выражение появится 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 пример на конструирование выражений. Очистим верхнее окно с помощью команды **Clear** (нажав клавиши Т и С). Теперь введем две строки с выражениями **sin(2*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 только команд главного меню и покажем, как можно вычислить численное значение выражения в строке 3 при X=1.5. Для этого воспользуемся позицией **Manage** (нажав клавишу М) и далее командой **Substitute** (нажав клавишу S). При этом надо указать номер строки вычисляемого выражения (в нашем случае п). Появится запрос

MANAGESUBSTITUTEvalue:<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 вместо переменной X в исходное выражение численное значение 1.5. Теперь, используя команду **approX** (нажатием клавиши X) и указав номер строки 4 с вычисляемым выражением, нажав клавишу **Enter** можно получить результат вычислений:

5: 0.212212

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

Уничтожение строк (Remove)

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

Для того чтобы убрать строки документа, содержащие ненужную информацию, служит команда уничтожения строк **Remove.** При ее вводе система выводит в нижней части экрана запрос

REMOVE: Start:n1 End:n2

По умолчанию в этом запросе указан начальный n1 и конечный п2 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 для последующих операций. Это справедливо, даже если уничтожаются все строки.

Автоматическая очистка оперативной памяти

Очистка ставших ненужными строк и ввод новых выражений ведут к заполнению памяти компьютера. Принято называть ставшие ненужными определения, хранящиеся в памяти, «мусором».

Следует также учитывать, что многие операции требуют временного занятия значительных объемов памяти. В нижней части экрана имеется сообщение вида Memory nn%, сообщающее о том, какой процент памяти компьютера занят. Надо следить, чтобы значения nn% не приближались к 0, так как в этом случае работа системы будет прекращена с выдачей сообщения о переполнении памяти.

Данные о проценте использования памяти обновляются только в моменты чистки памяти от «мусора». Необходимость в такой очистке Derive определяет автоматически. На время очистки памяти работа системы приостанавливается. Если чистка начинается в момент, когда происходит упрощение выражения, в строке состояния появляется значок прямоугольника. Для полной очистки системы без выхода из нее следует использовать команду **Clear** в позиции **Transfer** основного меню.

Задание глобальных опций (Options)

Опции отображения информации

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 т. д. При активной позиции **Options** появляется подменю, содержащее ряд команд. Их названия и выполняемые функции даны 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. Эти 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. Полезно, например, сделать фон окна выражений белым, а цвет символов меню и выражений черным (именно так они представлены в данной книге).

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 (Standart) 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.

Опции прямого исполнения команд

Execute — прямое исполнение команд MS-DOS. Можно, например, просмотреть каталог файлов, выполнить их переименование или стирание и т. д. Нажатие после этого 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.

Опции задания нотации и точности представления чисел

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"

о. $\dfrac{1000000}{3}\ x$

4: "Опция Notation Decimal"

5: 333333.3 x

6: "Опция Notation Mixed"

7: 333333.3 x

8: "Опция Notation Racional"

а· $\dfrac{1000000\ x}{3}$

10: "Опция Notation Scientific"

11: $3.33333\ 10^{5}\ x$

Precision — установка точности вычислений. Возможна установка следующих относящихся к этому режимов: приближенные вычисления — аппроксимация (Approximate), точные целочисленные вычисления (Exact) и смешанные (Mixed). Здесь также можно задать число верных знаков результата (Digit). Приведенный ниже пример показывает, как можно менять точность вычислений:

1: "Действие команды Precision"

2: "Исходное выражение"

3: $-\dfrac{40!}{3}$

4: "Установлена опция Approximate Digit:6"

5: $2.719717612088184295268763933614069853851870200644\ 10^{47}$

6: "Установлена опция Exact Digit:50"

7: $2.719717610826325781152037565320386314239999999999\ 10^{47}$

8: "Установлены опции Mixed и Decimal (Digit:50)"

9: 271971761082632578115203756532038631424000000000

10: "Новое выражение"

11: $\dfrac{1}{7}$

12: 0.14285714285714285714285714285714285714285714285714

Опции задания основания чисел

Radix — установка основания для числовых данных в пределах от 2 до 36, по умолчанию 10. Основания могут устанавливаться раздельно 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. В их числе двоичные, восьмеричные и шестнадцатеричные числа, широко используемые в вычислительной технике. Разряд шестнадцатеричных чисел представляется цифрами от 0 до 9 и затем латинскими буквами A, B, C, D, E и F.

Работа с внешними устройствами (Transfer)

Для работы с основными внешними устройствами — дисковыми накопителями и принтером — служит команда **Transfer.** С помощью этой команды и соответствующих подменю Derive позволяет записывать документы на диск и считывать их с диска, а также посылать их на распечатку принтером или в дисковый файл с особым (принтерным) форматом представления.

Команда **Transfer** порождает подменю, имеющее ряд позиций. Они и относящиеся к ним действия отмечены ниже:

Load — загрузка нового файла с расширением .mth с уничтожением в редакторе содержимого ранее загруженного 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 рассмотрены данные команды более подробно.

Работа с дисковыми накопителями

При подаче команды **Load** появляется дополнительное подменю:

TRANSFER LOAD:	Derive	State	daTa	Utility
Загрузка	Derive	Статус	Данные	Утилиты

Это подменю определяет тип загружаемого с диска файла. В позиции Derive загружаются тексты документов — файлы с расширением .mth. Эти тексты по мере загрузки выводятся на экран дисплея. Строки при этом в документе проставляются в строгом порядке, начиная с 1 и с шагом, равным 1.

Команда **State** загружает файл derive.ini, хранящий данные о состоянии системы, а команда **daTa** позволяет загружать данные (числа, векторы и матрицы), записанные в текстовом формате. Эта команда в первых версиях Derive отсутствовала. Наконец команда **Utility** также загружает файлы с расширением .mth с внешними расширениями (утилитами) системы. Однако при этом загружаемые документы на экран дисплея не выводятся и поступают в память компьютера. 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 для примера, рассмотренного выше, 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 Expande):"
        -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 Rational:"
        (x^2-5)/(x*(x-1)^4)

"Для #10 используем команду Factor Rational:"
        (x^2-5)/(x*(x-1)^4)
```

Файл derive.ini также является текстовым файлом и содержит многочисленные данные о состоянии системы на момент записи этого файла. Ниже приведено несколько первых строк этого файла:

PRECISION |Exact|

PRECISION-DIGITS 6

NOTATION |Rational|

NOTATION-DIGITS 6

INPUT-BASE 10

OUTPUT-BASE 10

INPUT-MODE |Character|

....

Заметим, что размеры распечаток документов в этом файле даются в знакоместах, а цвета заданы их стандартными кодами. Этот файл формируется системой, но его можно скорректировать и с помощью внешнего текстового редактора.

При работе с командой **Save** появляется подменю

TRANSFER SAVE: Derive Basic C Fortran Pascal Options State

Первые пять команд обеспечивают запись документов (по запросу их имени) в виде файла с форматом Derive, Basic, C, Fortran и Pascal. Это позволяет вводить выражения и результаты вычислений из документов Derive в программы, написанные на наиболее распространенных языках высокого уровня.

Команда **Options** позволяет записать опции системы. Она имеет подменю

TRANSFER SAVE OPTIONS: Range: All Some Annotation: (Save) Omit Length:79

Наконец команда **State** позволяет записать текущее состояние системы (по умолчанию в файл derive.ini, но можно создавать свои файлы инициализации системы с расширением .ini и затем загружать их командой **Load).**

Работа с принтером

Пункт основного меню **Print** имеет подменю

TRANSFER PRINT: Expression Screen File Layout Options

Команда **Expression** 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. Новая команда **Screen** позволяет копировать содержимое экрана.

Команда **Layout** позволяет установить размер страницы и полей:

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

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

Перемещение строк **(moVe)**

В процессе подготовки документов в системе Derive часто возникает необходимость в перемещении 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
(Перенести)	(Перед)	(Начало)	(Конец)

Enter label number:
(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 уничтожать заданный блок строк. С помощью этих команд можно навести должный порядок в документе перед его записью на диск.

Внимание

Derive под MS-DOS не имеет команды перенумерации строк после перемещения некоторых из них или удаления. Однако для перенумерации можно просто записать документ в виде файла командой ***Save,*** *а после этого считать записанный файл командой* ***Load.*** *Строки при этом будут перенумерованы в строго последовательном порядке.*

Работа с окнами (Window)

Система Derive позволяет разбить экран дисплея на несколько окон и выводить в каждое из них свою текстовую и графическую информацию. Для проведения с окнами различных операций используется позиция основного меню Window. Она имеет 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 из этих команд достаточно очевидно.

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)

После этого запроса 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 порождает подменю с запросами

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

Подробно операции с окнами будут описаны в уроке 5. К сожалению, Derive не имеет средств для записи состояния со многими окнами.

Управление окнами

Для управления окна с текстами документов используются следующие клавиши:

F1 — перейти к следующему окну;

F2 — переключить перекрывающиеся окна;

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.

Работа с помощью (Help)

Позиция **Help** основного меню вводит систему помощи. На экран дисплея выводится перечень 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 наименований позиций меню на русский язык дан ниже (в русифицированной Derive меню также имеет этот вид):

Справочник по Derive (Help)
E — команды строчного редактора
F — функции и константы
A — команды окна алгебры
2 — окно 2-мерной графики
3 — окно 3-мерной графики
U — функции файлов-утилит
S — текущее состояние системы
R — Возврат в Derive
Нажмите букву, соответствующую выбранному пункту

При обращении к справке в нижней части экрана появляется подменю помощи, разделы которого повторяют приведенные выше (рис. 2.14). Для ввода нужного раздела помощи достаточно сделать активной соответствующую позицию подменю и нажать клавишу **Enter.** На экране дисплея появится текст помощи на английском языке.

Система помощи в Derive достаточно проста и очень лаконична. Тем не менее она позволяет получить наиболее важные данные о системе и, главное, назначение встроенных операторов и функций, а также многих внешних расширений, поставляемых с системой в виде файлов с расширением **.mth.** На рис. 2.15 показан образец выдачи справки при нажатии клавиши E. При этом выдается справка по клавишам (и комбинациям клавиш), используемым для ввода и редактирования выражений строчным редактором.

При просмотре справки меню содержит три операции:
Next — переход к следующей странице справки;
Previous — переход к предыдущей странице справки;
Resume — переход к начальной странице справки (см. рис. 2.15).

Справочник **по Derive** (Help)

Б **-** команды строчного редактора
F - функции и константы
я - **команды** окна **алгебры**
2 - окно **2-мерной** графики
3 - окно 3-мерной графики
U - функции **файлов-утилит**
S - текущее состояние системы
Я - Возврат в **Derive**

Нажмите букву соответствующую выбранному пункту

HELP: **Editing** Functions **Algebra 2D-plot 3D-plot** Utility State Resune

Введите пункт меню

Free:100% **Ins** **Derive** Algebra

Рис. 2.14. Экран русифицированной версии Derive 4.08 под MS-DOS с окном справки

Пример вывода справки по константам и функциям системы дан на рис. 2.16. В тексте справок имеются указания на разделы описания системы, поставляемого при ее легальном получении.

Команды строчного редактора (п. 2.4)

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 неню
Ins **-** переключить рении **вставки/замены**
F1 - показать **короткую справку** (Help) по **командам,** функциям и т.д.
F3 - вставить выделенное выражение в строчный **редатор**
F4 **-** вставить выделенное **выражение,** заключив его в скобки
F6 - переключить решим
клавиш-стрелок(строчный_редактор/выделение_подвыражений)

Команды «правления курсором (в ретине выделения **выражений клавиши-стрелки** не **работают):**
Ctrl-S или <— - влево на символ t
Ctrl-D или —> - вправо на символ
Ctrl-A или Ctrl <— - влево на слова
Ctrl-F или Ctrl —> - вправо на слово
Ctrl-], Ctrl-Q S или Howe - к левому краю строки
Ctrl-S. **Ctrl-Q D** или **End** - к правому крап строки

HELP EDITING: **Next** Previous Resune

Введите пункт меню

Free:100% Ins Derive **Algebra**

Рис. 2.15. Образец выдачи справки по управляющим клавишам

```
                    Функции, константы и операторы

Логические операторы и функции(п.19.5)
  NOT p    истинно если p ложно
  p AND q  истинно если оба p и q истинны
  p OR q   истинна если p или q истинно
  p XOR q  истинно если p или q истинно, но не оба вместе
  p IMP q  истинно если p ложно или q истинно
  Если p и q - целые, та выполняются побитовые логические операции
  TRUTH_TABLE(p,q,...u,v,...)  дает таблицу истинности для переменных p,q>... и
логических выражений u,v,...

Решение уравнений и неравенств (п. 4.14):
  SOLVE (u,x) - решить уравнение u=0 относительно x
  SOLVE (u=v,x) - решить уравнение u=v относительно x
  SOLVE (u<v,x) - решить неравенство u<v относительно x
  SOLVE (u=v,x,a,b) - решить уравнение u=v относительна x в интервале [a,b], в
приближенном режиме
  SOLVE ([u1=v1,u2=v2,...], [x1,x2,...]) - решить систему, линейную относительно
x1, x2, ...

Экспоненциальные функции (п. 6.1):
  SQRT (г) - квадратный корень из z (или нажмите Alt-Q)
  EXP (z) - #e в степени z

HELP FUNCTIONS: Next Previous Resume

Введите пункт меню
                                        Free:100% Ins        Derive Algebra
```

Рис. 2.16. Пример выдачи справки по константам и функциям Derive 4.08

Итак, справочная система Derive под Windows весьма кратка и лаконична. Как серьезный ее недостаток надо отметить отсутствие примеров на каждый оператор и функцию. Тем не менее справочная система выполняет свою основную функцию — знакомство пользователя со средствами системы Derive под MS-DOS. Для более полного знакомства с возможностями системы читатель может обращаться к данной книге или к техническому описанию системы (увы, у нас крайне дефицитному).

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. С этой целью используются клавиши перемещения курсора вверх и вниз. Каждое их нажатие ведет к перемещению на одну строку.

Однако при работе с большими (многострочными) документами поиск нужной строки таким образом 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 клавишу Enter. Появится страница документа с активной указанной строкой.

Подсказки и индикация ошибок

Задание подсказок

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 символьные 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.

Внимание

Если пользователь при управлении системой ***нажимает*** *не, порождающие действий клавиши* ***или сочетания*** *клавиш, то Derive подает сигнал ошибки в виде короткого звукового гудка. Его можно* ***отключить*** *соответствующей командой* ***Options Mute.***

Сообщения об ошибках

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

Сообщение	Перевод сообщения
Zero Divide	Деление на ноль (результат **inf)**
Undefined Functuon	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
Memorry Full	Исчерпана память ОЗУ
Disk **Ful**	Диск заполнен
End_Of_File	Конец файла
Console Interupt	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 верным, 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.

Малое число сообщений об ошибках связано с двумя факторами. Во-первых, многие ситуации, которые в других (не символьных) системах ошибочны, в Derive будут корректными. 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 это 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, не работающих с верхней областью памяти. Поскольку профессиональные версии XM работают с памятью большого объема (до 4 Гбайт), то возникновение такого сообщения у современных ПК с реальным объемом ОЗУ в 16—32 Мбайта и выше маловероятно, хотя и не исключено при решении сложных задач: ведущих к известной в символьной математике проблеме разбухания результатов вычислений.

Выход из системы (Quit)

Если необходимо временно выйти из системы Derive в MS-DOS, то для этого 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 сохранят ются.

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 выйти 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 утеряны. Поэтому если они представляют ценность, то следует позаботиться об их сохранении (см. описание команд позиции **Transfer** основного меню). Для этого следует ответить на запрос N, использовать команду **Save** в позиции основного меню **Transfer** и уже затем выйти из системы.

Чему мы научились:

- Работать с Derive под MS-DOS.
- Использовать меню систем Derive под MS-DOS.
- Редактировать и вводить выражений.
- Строить математические выражения.
- Уничтожать и перемещать строки в сессии.
- Задавать глобальные опции системы.
- Работать с внешними устройствами.
- Работа с окнами и системой помощи.
- Пользоваться подсказками и сообщениями об ошибках.
- Выходить из системы при завершении работы.

Урок 3. Вычисления в командном режиме (Calculus)

- Обзор видов вычислений в командном режиме
- Вычисление производных
- Интегрирование выражений
- Вычисление пределов функций
- Вычисление произведения членов ряда
- Вычисление суммы членов ряда
- Разложение функции в ряд Тейлора

Обзор видов вычислений в командном режиме

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

Differenciate — 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 из 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.

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 Differenciate. По запросу

CALCULUS DIFFERENCIATE expression: #n <Выражение>

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

CALCULUS DIFFERENCIATE 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 DIFFERENCIATE Order: 1 <Порядок>

нужно указать порядок производной — 3 вместо 1 по умолчанию.

После этих действий во второй строке появится выражение с описанием производной. Далее, воспользовавшись командой **approX,** найдем значение третьей производной в символьном виде: -8 COS (2x). Документ, полученный в результате этих операций, представлен ниже:

```
1:      SIN (2 x)

        / d \3
2:      |—-|  SIN {2 x)
        \ dx /

3:      -8COS (2 x)
```

Рассмотрим еще ряд практических примеров на численное дифференцирование:

```
1: "Символьное дифференцирование — используется команда Simplify"

2: "Вычисление первой производной функции а^х"

        d      x
3:————————a
        dx

          x
4:      a  LN(a)

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

        d
6:—————————ASINH <x)
        dx

               1
        ——————————
7:                 2
        SQRT (x  +1)

8: "Вычисление третьей производной функции x^a"

        / d \3  a
9:      |—-|  x
        \dx/
```

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

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: $\dfrac{SIN\ (x)}{X}$

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

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

5: "Для вычисления численного значения выражения #4"
6: "используйте команду Substitute в позиции Manage"

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

8: "Затем используйте команду approX"

9: $\dfrac{1207}{7426}$

В примере 1.6 вычисляется численное значение производной. Для этого используется команда подстановки **Substitute** в позиции **Manage** главного меню. После замены x на численное значение (x = 0.5) используется команда вычисления **approX.**

Интегрирование выражений

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

CALCULAS INTEGRATE expression: #n <Номер выражения>
CALCULAS INTEGRATE variable: x <Переменная>
CALCULAS 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: $$\int \frac{\mathrm{SIN}^4(x)}{\mathrm{COS}(x)}\,dx$$

5: $$\mathrm{LN}\left(\frac{\mathrm{SIN}(x)+1}{\mathrm{COS}(x)}\right)-\frac{\mathrm{SIN}^3(x)}{3}-\mathrm{SIN}(x)$$

6: "Вычисление интеграла функции x*ln(a+b*x)"

7: $$\int x\,\mathrm{LN}(a+b\,x)\,dx$$

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

9: "Вычисление двойного интеграла"

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

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

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: $$\int_0^{inf} x\ EXP(-x)\ dx$$

22: 1

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

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

25: 2

26: "Неберущиеся интегралы"

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

```
          /  SIN (x)
28:       |————————dx
          /    x

          /
29:       / EXP (x) LOG (x) dx

                        /    x
             x          |  #e
30:       #e  LN (x) -  |———————dx
                        /   x
```

В строках 13—19 задано вычисление так называемого интеграла свертки, широко применяемого в электротехнических расчетах. А в строках 20—25 даны примеры на вычисление 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 не берущиеся интегралы, которые не выражаются в элементарных функциях. Примером может служить интеграл, вычисляемый в строках 27 и 28. В этом случае Derive отказалась взять интеграл и просто повторила его. Однако в случае, показанном в строках 29 и 30, интеграл представлен в виде, содержащем другой (тоже не берущийся) интеграл. Могут возникнуть трудности и с взятием интегралов от произвольных дробно-рациональных функций. В таких случаях их полезно разложить на простые дроби.

Приведенные выше 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 любым принтером. При этом естественный знак интеграла (видимый на экране дисплея) изображается символически в виде

```
    /  /Ъ
/  |   | Упрощенные
/  /   | знаки интеграла
      /а
```

Заметим также, что сплошная черта деления изображается пунктирной чертой вида———, а квадратные скобки даются в стилизованном виде. На экране дисплея в графическом режиме работы Derive эти упрощения отсутствуют. Их ввод связан с необходимостью распечатки документов любым принтером.

Ниже дан пример на вычисление численных значений определенных интегралов:

```
1: "Вычисление определенного интеграла"

2:        SQRT (2 x + 1)
```

```
              1
            /
3:         /   SQRT (2x + 1) dx
             0

           59984
4: ───────────────
           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 **approX** (что и сделано в конце приведенного примера).

Следует отметить, что далеко не всегда результат интегрирования даже простых выражений повторяет очевидный результат или результат, приводимый в справочниках. Однако, как правило, его можно привести к таковому путем более простых, чем операция интегрирования преобразований.

Зачастую причиной таких «казусов» являются свойства первообразной. Например, если проинтегрировать производную от выражения x/(x + 1), то Derive выдаст результат -1/(x + 1) вместо более очевидного x/(x + 1). Причина различий в том, что первообразные при интегрировании не являются единственными и могут отличаться на константу. В данном случае выражения 1/(x + 1) и **x/(x** + 1) отличаются на 1 и, таким образом, одинаково верно представляют **первообразную.**

Вычисление пределов функций

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

```
CALCULAS LIMIT expression: #n   <Номер выражения>
CALCULAS LIMIT variable: x   <Переменная>
CALCULAS 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 представлены примеры на вычисление пределов функций:

```
1: "Вычисление пределов функций"

           LN (1 + x)
2:     ───────────────
               x
```

3: $\lim\limits_{x \to 0} \dfrac{LN(1 + x)}{x}$

4: 1

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

6: $\lim\limits_{x \to 0} \dfrac{1 - COS(x)}{x^2}$

7: 0.5

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

9: $\lim\limits_{x \to 1-} \dfrac{2x - 2}{x - 1}$

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.

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:

CALCULAS PRODUCT expression: #n <Номер выражения>
CALCULAS PRODUCT variable: x <Переменная>
CALCULAS 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. Следующие примеры иллюстрируют это:

1: "Вычисление произведений членов ряда"

2: n

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

4: $n!$

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

6: $3.6288\ 10^{6}$

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

Вычисление суммы членов ряда

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

CALCULAS SUM expression: #n <Номер выражения>
CALCULAS SUM variable: x <Переменная>
CALCULAS 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 показаны примеры на вычисление суммы членов ряда:

1: "Вычисление сумм рядов"

2: $a\ q^{n}$

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

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

5: $2n - 1$

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

7: n^2

8: x^2

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

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

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

12: 55

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

В файлах **.prt** с текстами документов обозначения произведения П и суммы Ф заменены словами PROD и SUM, поскольку многие типы принтеров не имеют указанных знаков. Однако в данном случае мы привели распечатки с этими символами как примеры изображения, видимого на экране дисплея.

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 заяросы системы:

```
CALCULAS TAYLOR expression: #n   <Номер выражения>
CALCULAS TAYLOR variable: x   <Переменная>
CALCULAS 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 показывают следующие примеры:

1: "Разложение функций в ряд Тейлора"

2: COS (x)

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

4: $\dfrac{x^{4}}{24} \quad \dfrac{x^{2}}{2} + 1$

5: LN {1 + x)

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

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

8: $\dfrac{X^{5}}{5} \quad \dfrac{X^{4}}{4} + \dfrac{X^{3}}{3} \quad \dfrac{X^{2}}{2} + x$

В этом примере даны различные формы представления ряда Тейлора. Они зависят от того, какая из команд **(Simplify** или **approX)** используется для вычислений. Команда **Simplify** обычно задает вывод результатов вычислений в дробно-рациональном виде, а **approX,** 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. Рекомендуется просмотреть множество других примеров вычислений, описанных в гл. 4.

Чему мы научились:

- Выполнять вычисления в командном режиме.
- Вычислять производные.
- Вычислять интегралы.
- Вычислять пределы функций.
- Вычислять произведения членов ряда.
- Вычислять суммы членов ряда.
- Разлагать функции в ряд Тейлора.

Урок 4. Основы символьных вычислений

- Декларация новых определений
- Декларация переменных
- Декларация матриц и векторов
- Раскрытие выражений
- Синтез полинома по его действительным и комплексным корням
- Факторизация
- Решение уравнений
- Управление вычислениями и подстановки
- Упрощение выражений
- Команда численных вычислений

Декларация новых определений (Declare)

Derive позволяет задавать (декларировать) следующие новые определения: функции пользователя, переменные, матрицы и векторы. Для этого используется позиция главного меню **Declare,** имеющая подменю

DECLARE: Function Variable Matrix vectoR

Ниже дано описание этих деклараций с соответствующими примерами.

Декларация функций

Позиция **Function** подменю используется для задания функций пользователя, которые являются расширениями базового входного языка системы. Такие функции, после их задания, можно использовать в математических выражениях наряду с 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 сделать позицию **Function** активной, то появится запрос

DECLARE FUNCTION name:

В ответ на него надо ввести имя функции. После этого появится запрос

DECLARE FUNCTION value:

В ответ на него следует ввести математическое выражение, описывающее функцию. Пусть, к примеру, надо задать функцию трех переменных *f(x,y,z) = x + y * г.* 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 большими буквами (хотя задаваться может малыми). Список аргументов (в скобках после имени функции) формируется автоматически. В качестве знака присваивания используются символы :=, поскольку знак равенства = используется для иных целей (задания условий).

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), не отрицательными значениями (Nonnegative), с реальными значениями **(Real),** комплексными (Complex) и с интервальными (Interval).

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 раздела слов **_,** 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 буквами.

В отличие от первых версий Derive, в версиях 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 везде не меняется, а значение переменной можно всегда 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 **Autor)** 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!

Декларация матриц и векторов

Позиция **Matrix** подменю позволяет задать двухмерный числовой массив — матрицу. Как известно [15], 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 иметь как численные, так и символьные элементы матриц. Например, матрица с размерностью 2*3

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

это матрица с численными элементами, тогда как матрица

```
a    a+b   a-b
a*b  a/b   b
```

это матрица с символьными элементами.

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

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

DECLARE MATRIX: Rows:3 **Columns:3**

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

MATRIXelement: 0

нужно последовательно ввести значения всех элементов матрицы.

Позиция **vectoR** подменю служит для задания вектора — матрицы вырожденной в одну строку. При активной позиции **vectoR** появляется запрос

DECLARE VECTOR: Dimension

В ответ на него 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 |1 23|
|456|

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

[И]. [2]. [3]] задает вектор-столбец |1|
|2|
|3|

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.

Раскрытие выражений (Expand)

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 позволяет зачастую получать различные результаты преобразований.

В явном виде эта 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 данной команды она будет просто повторена в окне документов.

После пуска этой команды появляется запрос

EXPAND expression: #n

Надо указать номер строки выражения п, подвергаемого раскрытию. После этого 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. Пример на преобразование выражений в символьном виде дан ниже:

```
1: "Действие команды Expand"

2: "Исходное алгебраическое выражение"

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

4: "Результат его преобразования"

          2    2
5:       a  - b

6: "Преобразование тригонометрических выражений"

          SIN (x)
y, ______________

          COS (x)

8:      TAN (x)

                2          2
9:      SIN (x)  + COS (x)
10:      1

11:    SIN (a) COS (b) - COS (a) SIN (b)
```

12:"Этотрезультатвыполучите,толькоиспользуя"

13: "опции Collect и Trigonom позиции Manage"

14: SIN (a - b)

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

Синтез полинома по его действительным и комплексным корням

Интересным примером 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: $x^3 - 6 x^2 + 11 x - 6$

9: "Синтез полинома с комплексными корнями"

10: (x + 1 + #i 2) (x + 1 - #i 2) (x + 3) (x + 4)

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

Из приведенных примеров видно, что команда **Expand** является мощным средством для проведения символьных преобразований. Особенно удобна она при работе с многочленами и дробно-рациональными функциями.

Факторизация (Factor)

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*3*3607*3803.

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: Amound: Trivial Squarefree Rational raDical Complex

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

Область факторизации существенно влияет на возможность разложения **того** или иного **выражения.** Ниже представлены примеры на разложение 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: $-(-x - 1 + 2\#i)(x + 1 + 2\#i)(x + 3)(x + 4)$

В первом случае, строки 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: $a\ \mathrm{SINH}(x)$

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

4: $\mathrm{TANH}(x)$

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

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.** В этом случае команда **Factor** не работает ни при каких установках области факторизации. Однако в данном случае можно использовать команду расширения **Expand** — результат дан в строке 13.

Решение уравнений (soLve)

Решение нелинейных уравнений и неравенств

Команда **soLve** главного меню используется для решения уравнения вида **f(x)=0** или, в более общем случае **f1(x)=f2(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 решаемое уравнение.

Ниже представлены примеры на решения 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 approX, находим"
9:        x = -0.424031
```

Решение неравенств

Следует отметить, что команда **soLve** может использоваться для решения и неравенств. Результатом решения обычно является также неравенство. Это иллюстрируют следующие примеры:

```
1: "Решение неравенств — команда soLve"
2:        2 < EXP(x) < 3
```

```
3:      - SQRT(LN(3)) < x < SQRT(LN(3)) AND (x > SQRT(LN(2)) OR
         x < - SQRT(LN(2)))

4:      a < c*x < b

                      a                             b
5:      x*SIGN(c) >------AND x*SIGN(c) <------
                     |c|                           |c|

6:      2 < SQRT(x) < 10

7:      x > 4 AND 0 < x < 100
```

Вычисление действительных и комплексных корней полиномов

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

```
1: "Вычисление корней полинома — команда soLve"

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

3:      x = -2

4:      x = - SQRT (3) - 2

5:      x = SQRT (3) - 2

6:      x = -1 - 3 #i

7:      x = -1 + 3 #i
```

Как видно из этого примера, команда **soLve** позволяет найти как действительные, так и комплексные корни полинома. При этом надо установить комплексный формат представления чисел.

Решение систем линейных уравнений

Команда **soLve** может использоваться и для решения неравенств и систем уравнений, линейных относительно переменных, для которых ищется решение:

```
1: "Пример решения системы из двух линейных уравнений"

2:      [a*x + b*y = c, d*x + e*y = f]

            / c*e - b*f       a*f - c*d \
3:      x = |------------, y = ------------|
            \ a*e - b*d       a*e - b*d /
```

Решение уравнений в численном виде

Функция **soLve** может использоваться и для численного решения уравнений. Ограничимся двумя примерами решения одного и того же уравнения:

1: "Решение нелинейного уравнения в численном виде"

2: (EXP(x)) = 2

3: x = 2*LN(2)

4: SOLVE(i(EXP(x)) = 2, x, 0, 5)

5: [x = 2*LN(2)]

6: [x = 1.386294297]

Обратите внимание на функцию **SOLVE** в строке 4. Она дана в виде, позволяющем локализовать корень в заданном интервале — в нашем случае от О до 5. В таком виде обычно ищутся численные решения. Тем не менее приведенный пример показывает, что для получения привычного численного решения в форме вещественного числа приходится использовать функцию **approX** — так получено выражение в строке 5. Более подробно о технике такого использования команды **soLve** будет рассказано в дальнейшем.

Управление вычислениями и подстановки (Manage)

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 ему надо действовать. Для этого служит позиция **Manage** главного меню. При активной позиции **Manage** появляется подменю, содержащее следующие опции:

Branch — расширение по всем действительным корням;

Exponencial — 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 данном случае включены тригонометрические преобразования) иллюстрирует следующий пример:

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 (5 a)

11: "Используйте Manage Trigonometry Expand Sines"

12: $16\ \mathrm{SIN}\ (a)^{5} - 20\ \mathrm{SIN}\ (a)^{3} + 5\ \mathrm{SIN}\ (a)$

13: "Теперь используйте Manage Trigonometry Collect"

14: SIN (5 a)

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

Подстановки

Одна из самых важных опций в позиции **Manage** это опция-команда **Substitute (подстановка)**. Она используется для подстановки на место выбранной переменной или фрагмента выражения другого выражения. Следующий пример показывает 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: $1^3 - 6\ 1^2 + 11\ 1 - 6$

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.

Упрощение выражений (Simplify)

Команда Simplify — одна из самых важных команд системы Derive. В общем случае она выполняет упрощение символьных выражений (см. в гл. 4 более подробное пояснение функций этой команды). Применительно к числовым выражениям команда Simplify позволяет представить результат вычислений в виде рациональных чисел. Ниже дается пример на применение команды Simplify:

1: "Работа с командой Simplify:"

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

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

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

```
              x + a
         ——————————————
5:             2
         x (x  + c) (x - b)

                     2                        2
6:       (x + a) SIN (x)  + (x + a) COS (x)  - (x + a) + 1
7:       1
```

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

Команда численных вычислений (appгоX)

Команда **appгоX** (от слова approximate — аппроксимация) используется для приближенных вычислений математических выражений. Слово «при**ближенных»** не следует воспринимать в смысле «грубых» или «неточных». В действительности алгоритм вычислений гарантирует правильность всех цифр результата, за исключением последней в дробных числах.

При использовании команды **appгоX** появляется запрос

APPROXIMATE expression #n

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

Чему мы научились:

- Декларировать новые определения и переменные.
- Декларировать матрицы и векторы.
- Раскрывать выражения.
- Синтезировать полиномы по их корням.
- Выполнять разложение (факторизацию).
- Решать уравнения.
- Управлять вычислениями и подстановками.'
- Упрощать выражения.
- Приближенно вычислять выражения.

Урок 5. Графика систем Derive под MS-DOS (Plot)

- Построение двухмерных графиков в декартовой системе координат
- Построение графика параметрически заданных функций
- Построение графика функций в полярной системе координат
- Построение **3D-графиков** функций двух переменных
- Построение графиков специального вида

Построение двухмерных графиков в декартовой системе координат

Derive под MS-DOS 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 (двухмерная **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 двухмерной графикой Derive позволяет на одном экране строить несколько графиков для различных функций. Если это допускают возможности дисплея, то графики разных функций выделяются разными цветами. Derive строит графики в два захода. Вначале редкими точками быстро выполняется грубое построение графика, а затем уже он строится 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 на клавишу 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 двухмерных функций имеется 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:

PLOT:	Beside	Under	Overlay
	Рядом	Ниже	С перекрытием

По умолчанию задана позиция **Beside.** При этом 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 — дает возврат к главному меню и заданию функции;

Center — изменение положения центра координатной системы (он занимает такое положение, при котором графический курсор оказывается в центре графика);

Delete — стирание графика или графиков (команда имеет опции: **All** — стирание всех графиков; **Butlast** — стирание предыдущего графика; **First** — стирание последнего графика);

Help — включение системы помощи;

Move — быстрая установка курсора по заданным его координатам;

Options — установка опций 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; **State** — установка типа координатной системы Rectanqular и Polar, т. е. прямоугольная или полярная);

Plot — построение очередного графика;

Quit — выход из Derive в MS-DOS с прекращением работы;

Scale — изменение масштаба (одного деления масштабной сетки из точек);

Ticks — изменение густоты масштабной сетки (с параметрами **Rows** — строки и **Columns** — столбцы, параметры задаются в знакоместах);

Window — управление окнами;

Zoom — изменение размеров окна вдвое по обоим осям координат (опция **Both)** или по осям X и Y (опции X, Y) в направлении внутрь графика или из него (по запросу Direction: In Out).

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

F1 — переключение на следующее окно;

F2 — переключение перекрывающихся окон;

F5 — переключение к предшествующему режиму 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;

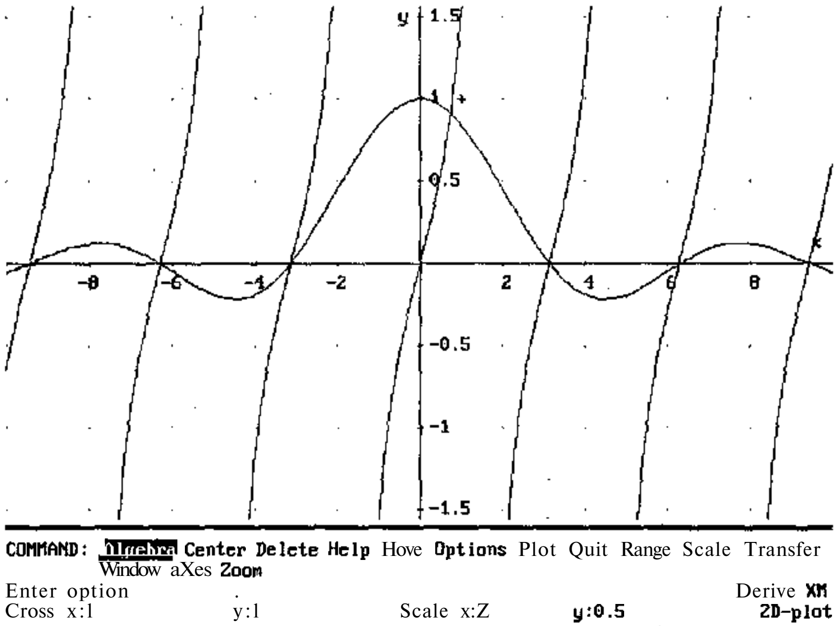


Рис. 5.1. Графики двух функций в декартовой системе координат

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 видно обычно ненужное для представления результатов меню системы.

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

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

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

Рассмотрим пример на 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: $\dfrac{\text{SIN (x)}}{\text{x}}$

3: TAN (x)

Далее, дважды используя команду **Plot** и установив Scale x=2 и x=.5, получим совмещенные графики этих функций (рис. 5.1).

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

Построение графика параметрически заданных функций

Теперь рассмотрим построение графика параметрически заданных функций. 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. 5.2. График образует фигуру Лиссажу.

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(theta).

График представляет собой линию, которую описывает конец вращающегося радиус-вектора **r** с началом в центре координат при изменении параметра — угла theta в заданных пределах (обычно от 0 до 2л).

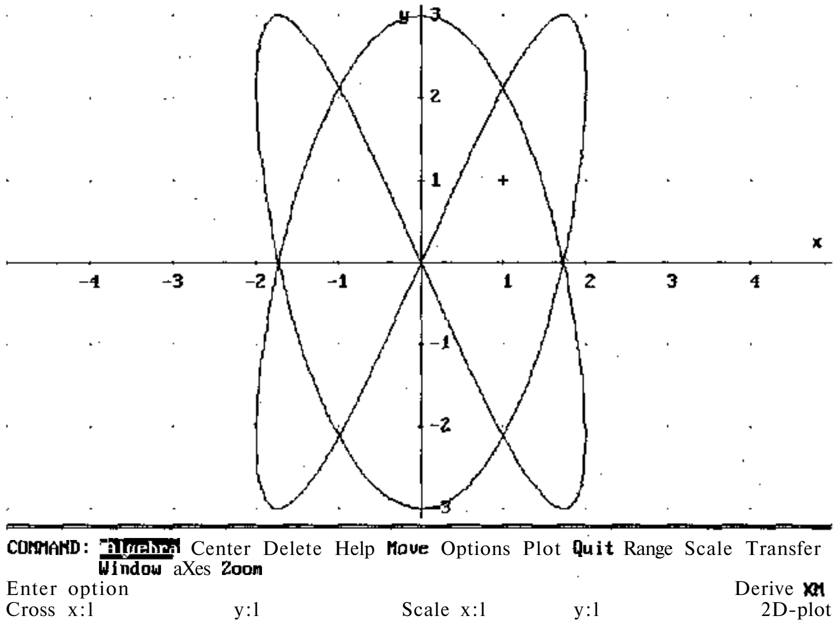


Рис. 5.2. График параметрически заданной функции

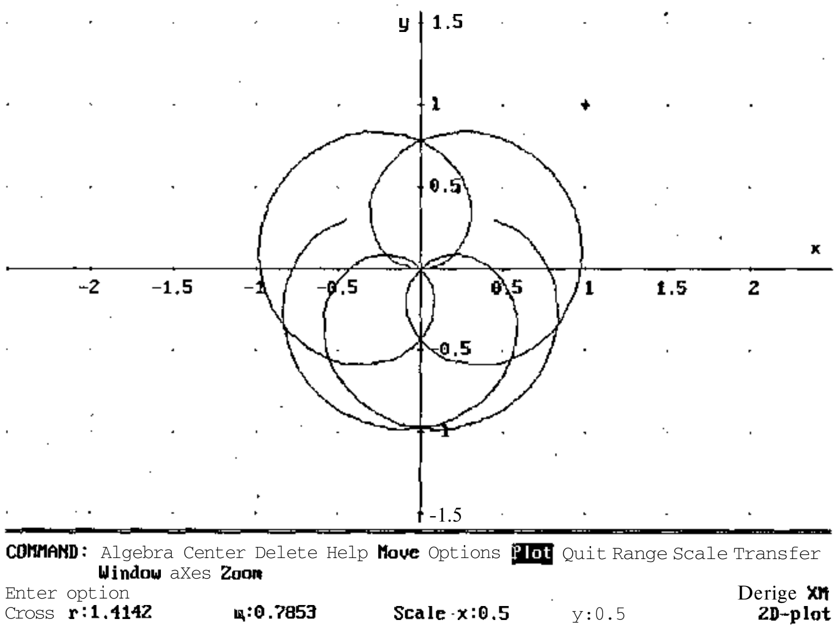


Рис. 5.3. График функции в полярной системе координат

Рассмотрим пример на построение графика функции в полярной системе координат. Пусть нужно построить график функции r = sin(t*4/7). Для этого достаточно ввести правую часть этого выражения и использовать команду Plot. После перехода в графическое окно надо, используя команду **State Polar,** установить полярную систему координат. Далее, указав команду **Plot** и задав пределы изменения t и масштабные коэффициенты по осям x и y, можно построить график функции, показанный на рис. 5.3.

Представленный на рис. 5.3 график построен при изменении t от -10 до 10. По умолчанию этот диапазон задан от -я до π.

Построение **3D-графиков** функций двух переменных

Система Derive под MS-DOS имеет простейшие 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. Увы, функциональная окраска фигур при построении **3D-графиков** в Derive под MS-DOS не предусмотрена. Она появилась только в последней версии Derive 5.* под Windows.

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

Algebra	Center	Eye	Focal	Grids	Hide	Length
(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 Length) порождают подменю вида

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

Их параметрами являются размеры или координаты в виде чисел n1, п2 и п3. После переустановки параметров следует повторить команду Plot для построения графика.

Опция Grid порождает подменю

GRIDS: x:n1 y:п2

Она устанавливает число линий каркаса, в виде которого строится фигура, по осям x и y. Чем выше значения nl и п2, тем лучше прорабатываются детали фигуры, но построение идет дольше. Оптимальные значения nl и п2 можно уточнить лишь экспериментально (по умолчанию они равны 10).

Опция Hide также имеет подменю

HIDE: Lines: Yes No

В зависимости от установки будет включен или выключен алгоритм устранения невидимых линий. Если он выключен, то 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. При этом 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 следующий пример. Пусть нужно построить график следующей функции двух переменных:

z(x,y) := cos(0.04*(x^2+y^2)) (5.1)

Для этого 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 Grid зададим по 20 линий каркаса по осям x и y. Далее, используя команду Plot, получим график, показанный на рис. 5.4.

Теперь введем новую функцию в виде

cos((x^2+y^2)/4)/(3+x^2+y^2) (5.2)

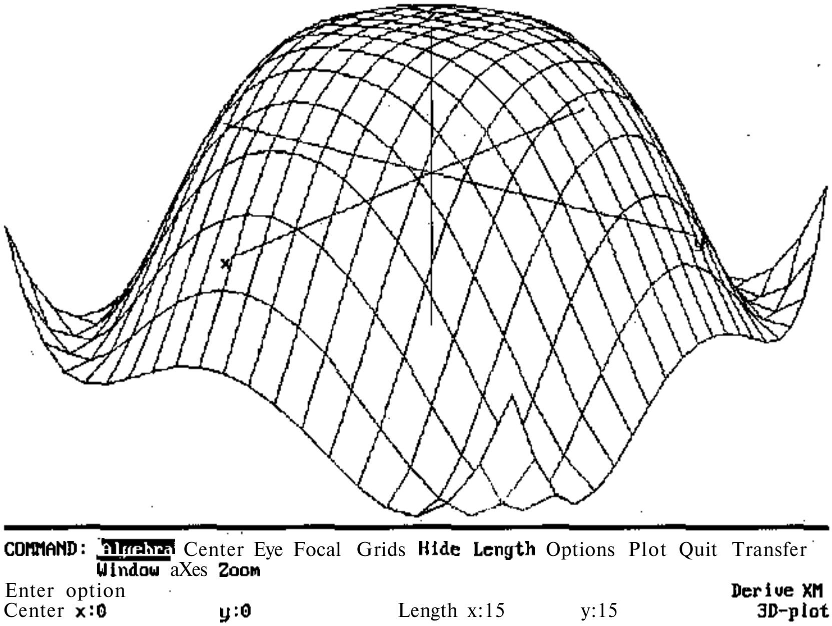


Рис. 5.4. График поверхности (5.1)

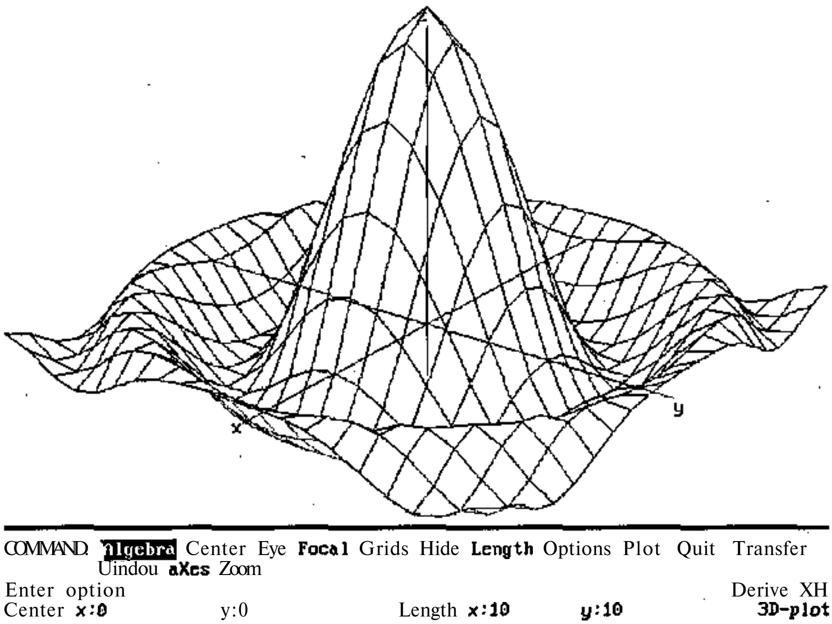


Рис. 5.5. График поверхности (5.2)

График этой функции (рис. 5.5) напоминает ковбойскую шляпу сомбреро с более острой вершиной, чем у фигуры, показанной на рис. 5.4.

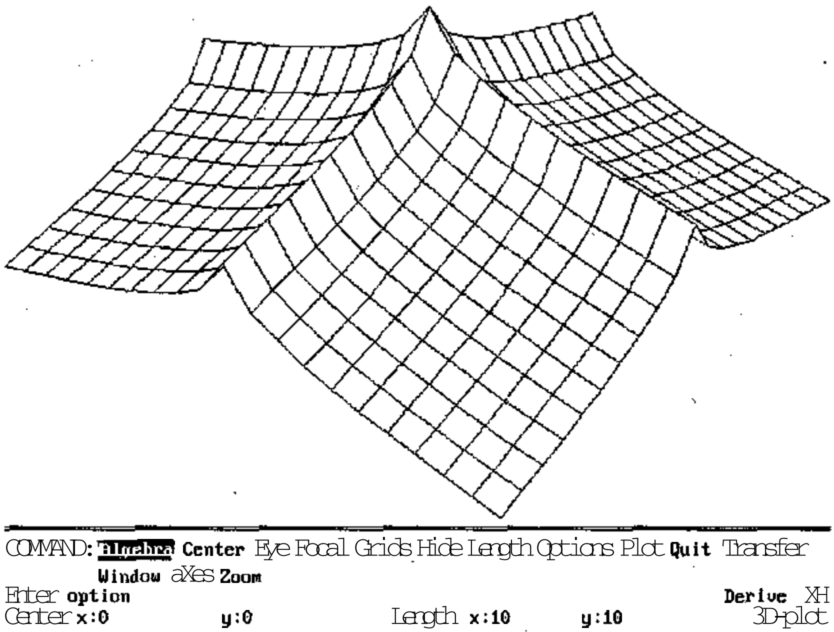


Рис. 5.6. График поверхности (5.3)

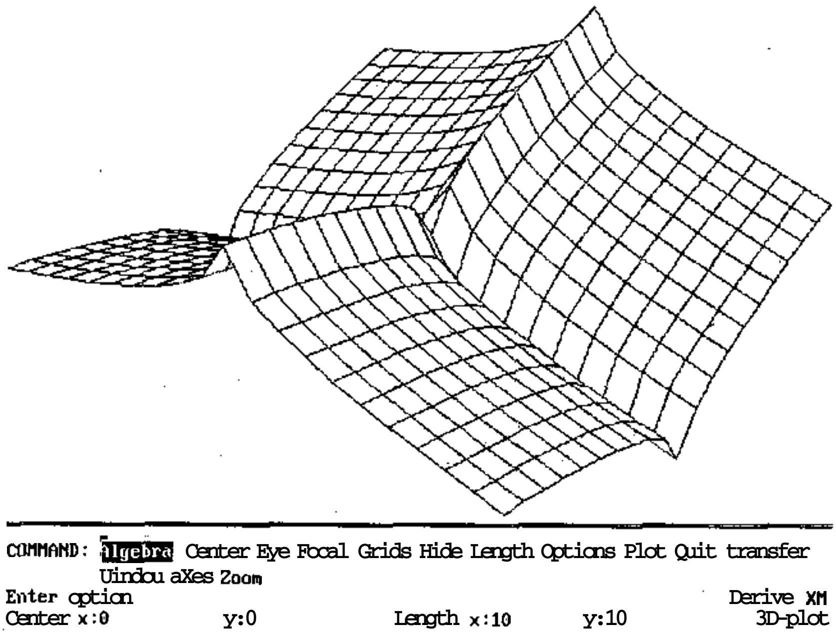


Рис. 5.7. График поверхности (5.4)

Далее построим график функции

-sqrt(abs(x)) – sqrt(abs(y)) (5.3)

Этот график показан на рис. 5.6 и очень напоминает фигуру, которую получают, деформируя четырьмя пальцами согнутый по перекрестным линиям сгиба лист бумаги.

И наконец, введем еще одну функцию:

sqrt(abs(x))-sqrt(abs(y)) (5.4)

Ее график, показанный на рис. 5.7, наглядно иллюстрирует, как резко меняется изображение даже при смене знака у одного из членов выражения, — только этим и отличается выражение (5.4) от (5.3). Фактически поверхность (5.4) обращена пиком вниз, тогда как (5.3) пиком вверх.

На рис. 5.4 и 5.5 видны координатные оси x,y и z, которые система строит по умолчанию. Однако с помощью опции **aXes** 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 некоторые возможности математической трехмерной графики Derive. Большое число примеров задания трехмерных фигур дается в файле **plot3d.mth,** примеры которого рекомендуется самостоятельно проработать.

К сожалению, ранние версии Derive под MS-DOS не предусматривали запись графиков на диск. Однако для этого можно использовать соответствующие резидентные программы из других графических пакетов. После загрузки такой резидентной программы, например **prtscr.com,** копию графика можно распечатать принтером, нажав одновременно клавиши **Shift** и **Print Screen** (на некоторых ПК только **Print Screen).**

Визуализация многих функций и их графиков

Используя команды работы с окнами, можно на одном графике представить запись арифметических выражений и графики различных функций. Покажем это на следующем примере. Пусть нужно построить одновременно графики четырех функций одной переменной x, расположив их и соответствующие расчетные выражения в 8 окнах. Для этого:

1. Используя команды **Window** и **Split,** разделим экран на 8 окон, по два окна по вертикали и по четыре окна по горизонтали еще на 4 окна.

2. Используя команду **Window** и **Goto 1,** перейдем в окно 1 (номер окна виден в его верхнем левом углу, активное окно помечено отличным цветом). Введем в это окно первое выражение (см. рис. 5.8), используя команду **Author.**

3. Используя команду **Window Goto 2,** 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. 5.8.

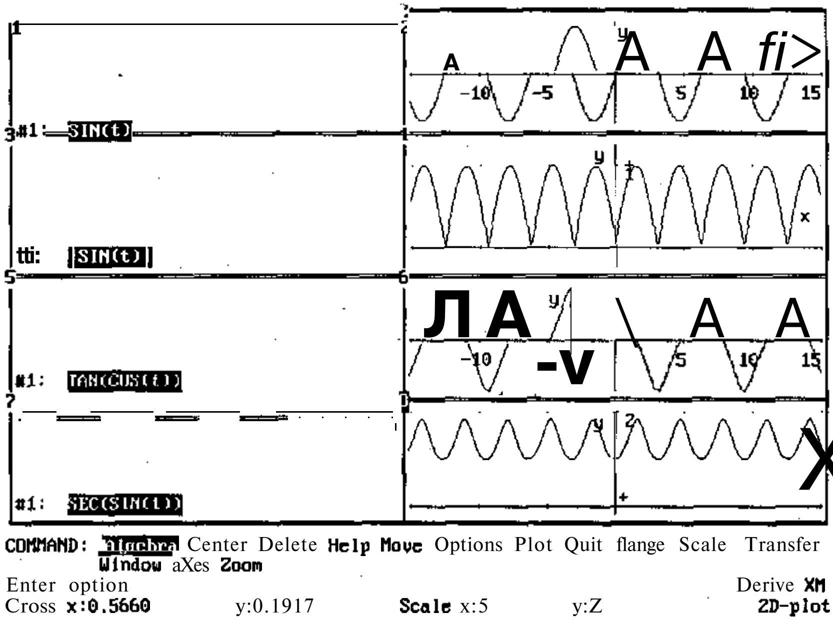


Рис. 5.8. Четыре функции и их графики на одном экране

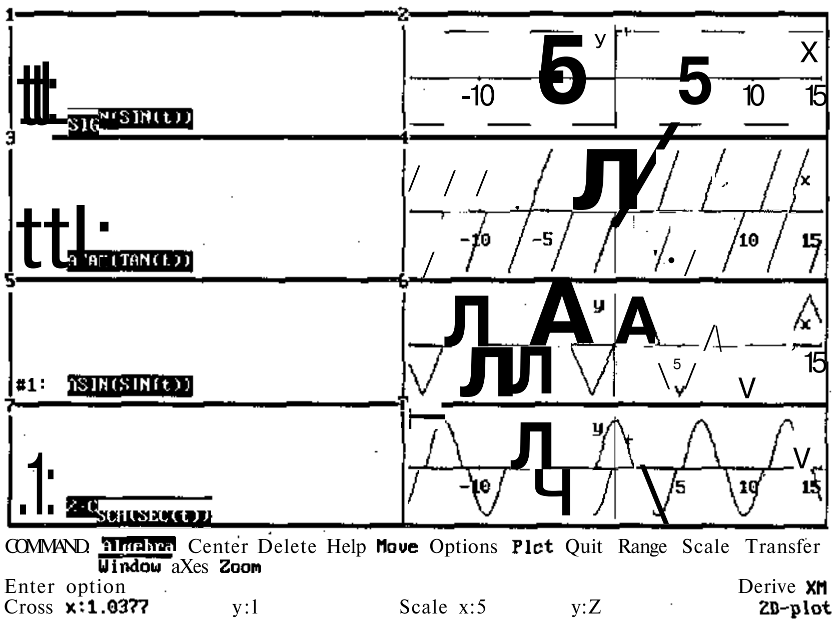


Рис. 5.9. Еще четыре функции и их графики на одном экране

Рис. 5.8 показывает, что использование многооконного режима существенно повышает наглядность представления как формульной, так и графической информации. Так, появляется возможность наглядно сопоставлять формы различных графических зависимостей, многие из которых весьма полезны для моделирования различных сигналов.

На рис. 5.9 показаны формулы и графики еще 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.

Визуализация ряда функций и графиков разного типа

Derive позволяет выводить в окна графики разных типов, например 2D в декартовой и полярной системах координат и 3D-графики. Это иллюстрирует следующий пример — создание четырех окон с формулами в одном из окон и графиками в других окнах.

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

1. Введем с помощью команды **Author** три выражения:

sin(x)/x	Функция одной переменной
[sin(3*w), cos(5*w)]	Функция, заданная в параметрической форме
exp(x^2+t^2)	Функция двух переменных

2. Разделим окно на два окна. Для этого используем команды **Window Split** и **Vertical** (напоминаем, что для этого достаточно нажимать клави-

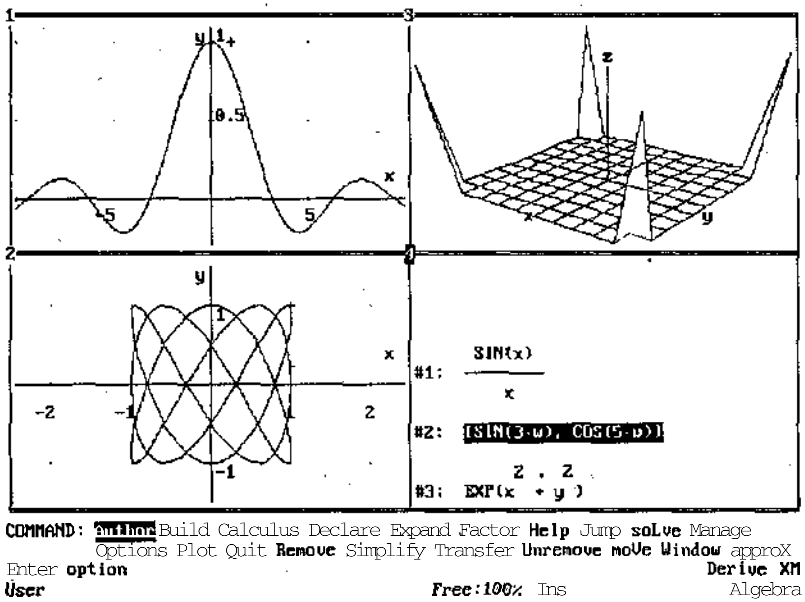


Рис. 5.10. Построение графинов трех разных типов на одном экране ши **W, S** и **V**). Согласившись с разделом окна по 40 столбцу, получим на экране два окна — левое и правое. Затем разделим их надвое по горизонтали. Указанные выражения будут во всех окнах.

3. Используя команды **Window** и **Goto 1,** сделаем окно 1 активным. Выделим в нем выражение #1. Затем используя команды **Window** и **Designate,** переопределим 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. Далее, 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** переходим в окно 3, т. е. делаем 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, строим график функции по выражению #3.

В результате этих построений получим рис. 5.10. Он дает наглядное представление о всех трех основных формах графических построений, реализованных в системе Derive. При этом в окне 4 дается запись самих функций в том же порядке, в каком даны их графики.

Построение графиков специального вида

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.

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(abs(F(#i*w)),10)].
```

Заметим, что при этом оператор p заменяется на #i*w, где w — круговая частота, и вычисляется логарифм модуля передаточной характеристики. Для построения диаграммы Бодэ достаточно ввести выражение

```
[re(F(#i*w), im(F(#i*w)]
```

Разбив экран на 3 окна, введя эти выражения в большое окно и построив (как описывалось выше) графики функций по двум последним выражениям, получим документ в виде рис. **5.11**.

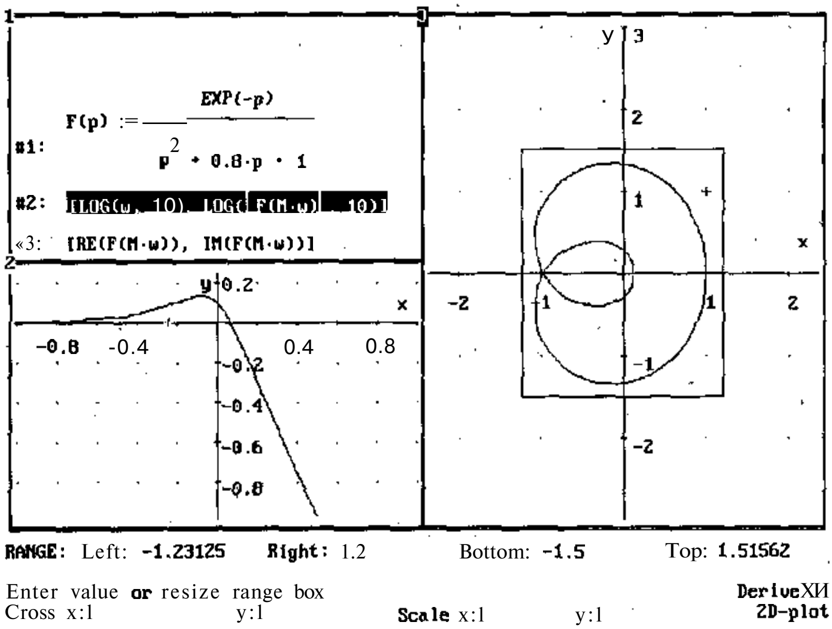


Рис. 5.11. Исходные формулы, ЛАЧХ и диаграмма Бодэ

График диаграммы Бодэ на рис. 5.11 показан несколько недостроенным. Это иллюстрирует весьма полезный способ выделения нужной части графика для его последующего построения в укрупненном масштабе. Состояние графика рис. 5.11 в окне #3 соответствует установкам пределов изображения (опций **Windows Range).** Изменение одного из параметров ограничения графика (слева, справа, снизу и сверху) отображается на графике отрезками прямых, образующих прямоугольник. Положение его сторон задается либо явно (заданием чисел), либо перемещением графического курсора. По завершении установки пределов изображения строится график, выделенный прямоугольником (рис. 5.12).

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 принтером.

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 самостоятельно построить графики нескольких функций.

Следует отметить, что копии экранов вида рис. 5.8, 5.9, 5.10 и 5.12 можно рассматривать как полезные опорные образы для визуализации соответствующих математических понятий. На уроках в школах и в вузах такие опорные образы могут служить раздаточным материалом, поясняющим те или иные разделы лекционных курсов.

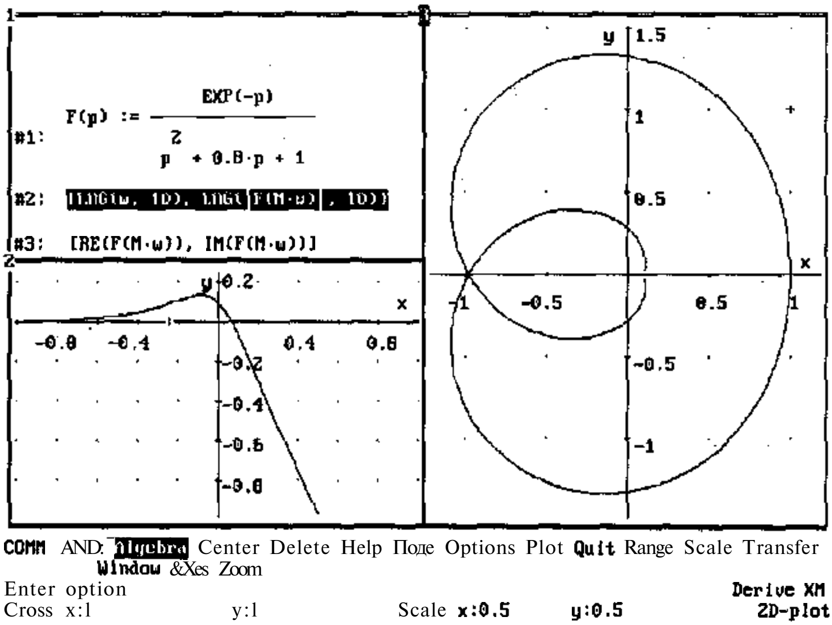


Рис. 5.12. Окончательно построенные окна с ИСХОДНЫМИ формулами и графики ЛАЧХ и диаграммы Бодэ

Чему мы научились:

- Строить двухмерные графики в декартовой системе координат.
- Строить графики параметрически заданных функций.
- Строить графики функций в полярной системе координат.
- Строить **3D-графики** функций двух переменных.
- Строить графики специального вида.

Урок 6. Работа с Derive 4 под Windows

- Начало работы с Derive 4 под Windows
- Строка меню и интерфейс пользователя
- Работа с файлами
- Редактирование выражений и документов
- Ввод данных
- Преобразование выражений
- Решение уравнений и неравенств
- Основные виды вычислений
- Объявление переменных и функций
- Подменю управления форматом ввода/вывода

Начало работы с Derive 4 под Windows

Начиная с версии Derive 4.0 под Windows система Derive стала типовым Windows-приложением. Имеется ряд' вариантов этой версии, отличающихся незначительно. Ниже мы рассмотрим хорошо известную у нас версию Derive 4.11 под Windows. В общем, мы будем называть семейство систем Derive под Windows первого поколения как Derive 4.* под Windows или даже про-

Рис. 6.1, Основное окно системы Derive 4 под Windows сто Derive 4 под Windows, поскольку рассмотрение деталей незначительных отличий в вариантах этих версий выходит за рамки данной книги и едва ли имеет большое значение.

При запуске Derive 4.* под Windows появляется основное окно, на фоне которого видна красочная заставка системы (рис. 6.1). В основном окне видны титульная строка, строка с главным меню и две панели: панель инструментов и внизу окна панель статуса системы.

Вычисления идут по схеме: «задал вопрос — получил ответ». Для ввода математических выражений предназначено диалоговое окно **Autor Expression** (рис. 6.2), которое открывается командой **Author • Expression.**

Рис. 6.2. Диалоговое окно ввода выражений

Если после ввода выражения щелкнуть на кнопке **OK,** выражение появится в окне выражений (рис. 6.3) и будет выделено цветом — белым на синем фоне. Если после ввода выражения щелкнуть на кнопке **Simplify,** выражение будет вычислено и в окне выражений появится результат. Щелчок на кнопке **Cancel** позволяет отказаться от операции и закрыть окно **Autor Expression.** Все введенные выражения и результаты их вычислений располагаются на отдельных строках и последовательно нумеруются символами вид #N, где N — номер строки.

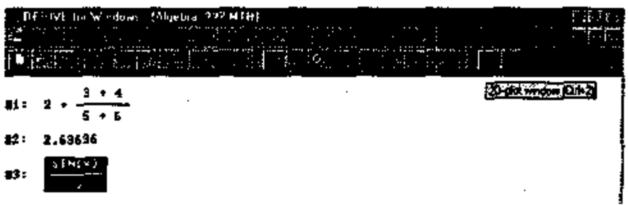


Рис. 6.3. Пример ввода и вычисления арифметического выражения

Для вычисления выражений в системе Derive имеется ряд других команд. Это команды **Basic** (результат в символьном виде) и **Approximate** (результат в численном виде) меню **Simplify.** При выборе этих команд появляются диалоговые окна, в которых достаточно щелкнуть на кнопке OK.

Рассмотрим пример на построение графика функции. В окне **Autor Expression** введите выражение SIN£x)/x (см. рис. 6.2) и щелкните на кнопке OK. На панели инструментов щелкните на кнопке **2D-plot windows (Ctrl+2)** (кнопка с изображением графика в декартовой системе координат), чтобы от-

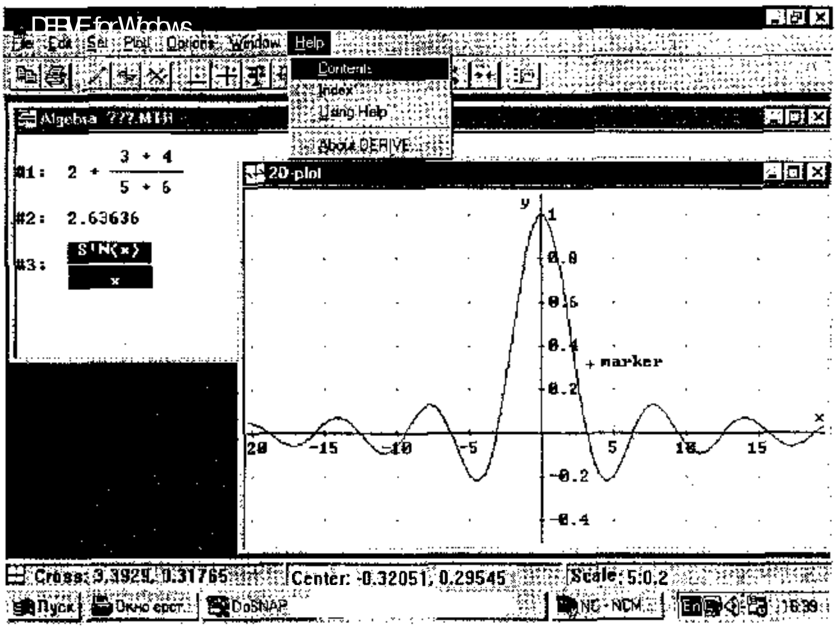


Рис. 6.4. Панель Derive с различными окнами

крыть окно, показанное на рис. 6.4 справа (в верхнем левом углу видно окно с введенными выражениями). Для построения графика надо вновь щелкнуть на кнопке **2D-plot windows (Ctrl+2)**, теперь уже в окне графика.

Оба окна (графика и выражений) можно перетаскивать, менять в размерах и располагать в любом порядке — сверху находится активное в данный момент окно графика. Активное окно выделяется темно-синим цветом строки заголовка. Чтобы сделать активным другое окно, щелкните на нем мышью.

Строка меню и интерфейс пользователя

Интерфейс пользователя системы Derive 4 под Windows позволяет выполнять большинство вычислений вообще без программирования, используя только команды меню. Строка меню, расположенная сверху, помимо обычных меню **File, Edit, Window** и **Help,** имеет следующие специфические:

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 — объявление функций и переменных;

Options — задание различных параметров системы.

Строка меню обеспечивает доступ ко всем командам системы. Набор позиций может модифицироваться в зависимости от текущего состояния системы. Некоторые команды меню при этом могут быть недоступными или вообще отсутствовать. На панели инструментов размещен ряд кнопок для быстрого доступа к некоторым командам. Они перечислены ниже (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 — вывод окна печати 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 указанного размера;
Author matrix — задание матриц указанного 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 двухмерной графики;
3D-plot window (Ctrl+3) — вывод окна трехмерной графики.

Применение кнопок на панели инструментов обеспечивает удобный, особенно для новичков, способ управления системой, однако они предоставляют доступ только к часто используемым командам, поэтому не стоит забывать о строке меню и «горячих» клавишах. Узнать название кнопки вам поможет всплывающая подсказка (см. всплывающую подсказку для кнопки **2D-plot window (Ctrl+2)** на рис. 6.3).

В нижней части окна Derive расположена строка состояния. В этой строке выводятся текстовые комментарии о состоянии 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. В окне выражений этого не видно.

Работа с файлами

Типы файлов в Derive могут быть различны — документы, библиотеки, демонстрационные файлы и т. д. Поэтому для загрузки файлов используется подменю **Load** со следующими командами:

Math — вызов диалогового окна загрузки документов с выражениями (расширение **.mth**);

Data — вызов диалогового окна загрузки файла с данными (расширение **.dat**);

Demo — вызов диалогового окна загрузки демонстрационных файлов;

Utility — вызов диалогового окна загрузки утилит.

На рис. 6.5 показан вид экрана в режиме просмотра демонстрационного файла. Нетрудно заметить, что в этом случае окно освобождается от второстепенных деталей и в левом верхнем углу появляются две кнопки с изображениями ладони — правая инициирует просмотр, а левая его останавливает.

Еще один важный тип файлов — утилиты, входящие в библиотеки расширения. Фактически это те же файлы с расширением .mth. Однако Derive имеет специальную команду **Utility** для их быстрой загрузки без отображения на экране дисплея.

В подменю **File,** помимо стандартных команд, имеются две характерные для Derive позиции: **Write To** — запись выражений в формате языков программирования и **Change Directory** — вывод диалогового окна для изменения текущей папки с файлами.

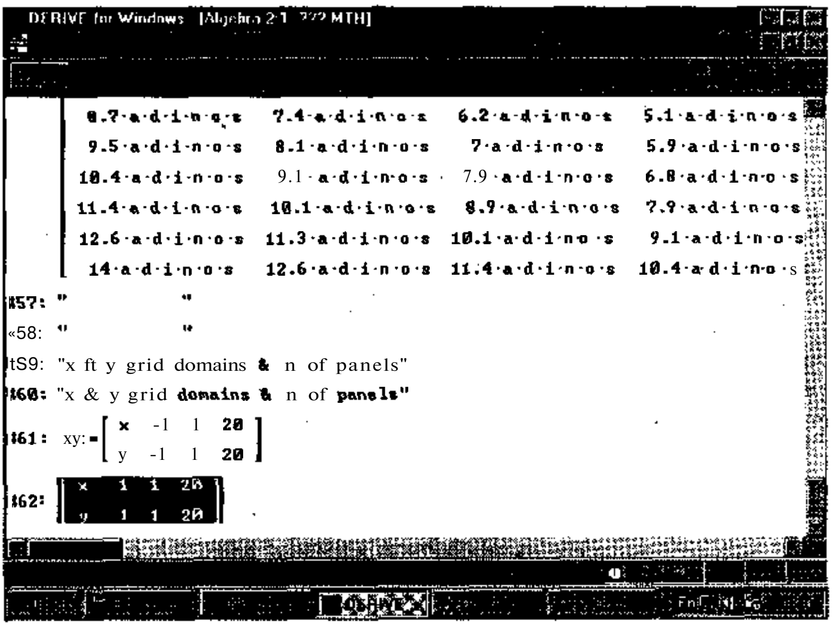


Рис. 6.5. Просмотр демонстрационного файла в системе Derive

В подменю **Write To** представлены следующие команды:

Basic file — запись выражений в формате файлов Basic;

C file — запись выражений в формате файлов C;

Fortran file — запись выражений в формате файлов Fortran;

Pascal file — запись выражений в формате Pascal.

Эти команды позволяют записывать выражения в файлы в формате популярных языков программирования Basic, C, Fortran и Pascal. Диалоговые окна, открывающиеся после выбора этих команд, практически аналогичны стандартному окну **Save As.** При использовании команд загрузки и записи файлов Derive выводит диалоговые окна с папкой, установленной по умолчанию. Чтобы изменить эту папку, достаточно воспользоваться командой **Change Directory.**

Редактирование выражений и документов

Команды редактирования

Основные команды редактирования сосредоточены в меню **Edit.** Ряд контекстно зависимых команд разбиты на три группы. В первой группе сосредоточены команды вывода окон редактирования выражений, комментариев и переходов к заданному выражению:

Expression — вывод окна редактирования выделенного выражения;

Annotation — вывод окна комментария для выделенного выражения;

Go to Expression — вывод окна для задания номера выражения, к которому требуется выполнить переход.

Если в окне выражений выделено какое-либо выражение цветом, команда **Expression** доступна, и ее выбор ведет к появлению описанного ранее окна **Autor Expression** (см. рис. 6.2). В нем имеется панель специальных символов и строка редактирования. Каждое выражение может снабжаться текстовым комментарием. Он вводится с помощью простого окна с полем ввода. Комментарий отображается в строке состояния, если выражение выделено.

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 диалогового окна для указания номера выражения.

Вторая группа команд служит для работы со строками документа:

Remove — удаление выделенного выражения;

Unremove — восстановление последнего удаленного выражения;

Move — перемещение выражения;

Renumber — перенумерация строки с выражением.

Команда **Remove** вызывает появление окна, в котором в полях Start# и End# можно ввести диапазон удаляемых строк. Для восстановления последних удаленных строк используется команда **Unremove.** Для перемещения строки или группы строк из одного места документа в другое помимо команды **Move** можно использовать метод перетаскивания (рис. 6.6).

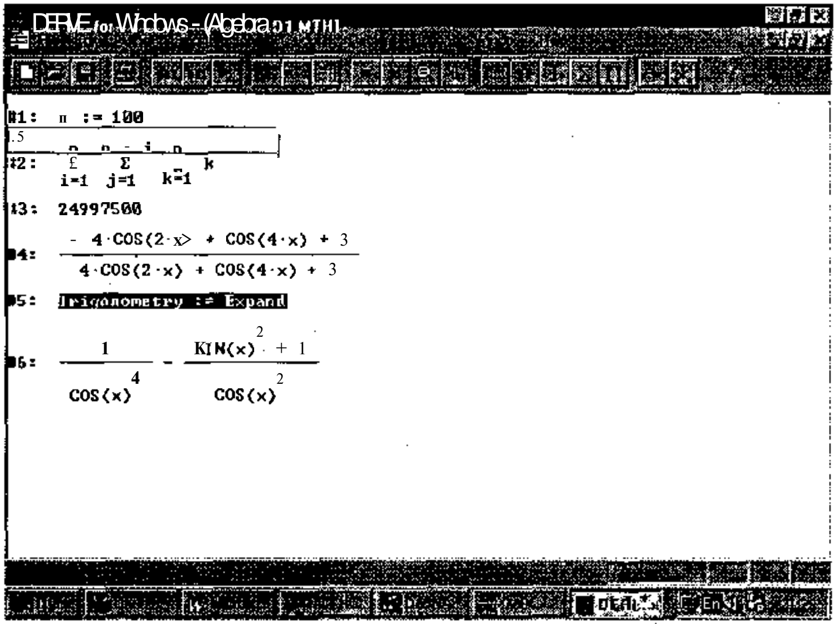


Рис. 6.6. Перетаскивание строки с выражением мышью

После перетаскивания строки с выражением (вместе с своим номером) автоматической перенумерации строк не происходит. Если вы заранее знаете, куда надо переместить блок строк, целесообразно использовать команду **Move.** Она вызывает появление диалогового окна с тремя полями ввода:

Before# — номер строки, перед которой размещается перемещаемый блок;

Start# — номер начальной строки блока;

End# — номер последней строки блока.

Например, если вы хотите перед строкой 5 поместить блок строк от 12 до 16, введите следующие значения: Before# — 5, Start# — 12 и End# — 16.

Для перемещения одной строки надо указать ее номер как в поле **Start#,** так и в поле **End#.** При использовании команды **Move** перенумерации строк также не происходит. Для этого используется команда **Renumber.**

Работа с буфером обмена

Команды последней группы служат для работы с буфером обмена:

Copy Expressions — копирование выделенного выражения в буфер обмена;

Mark and Copy — выделение и копирование выражения.

Любую выделенную строку с выражением или блок строк можно поместить в буфер обмена Windows, используя команду **Copy Expressions.** Номера строк при этом не копируются. Возможна передача содержимого буфера в другое приложение с помощью команды **Edit ▸ Paste** (рис. 6.7).

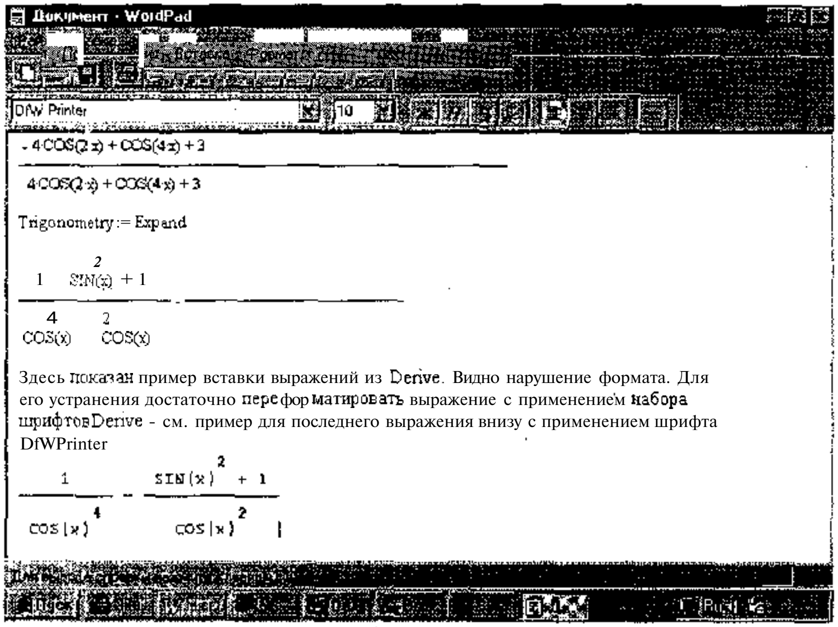


Рис. 6.7. Передача содержимого буфера в текстовый редактор WordPad и переформатирование выражений

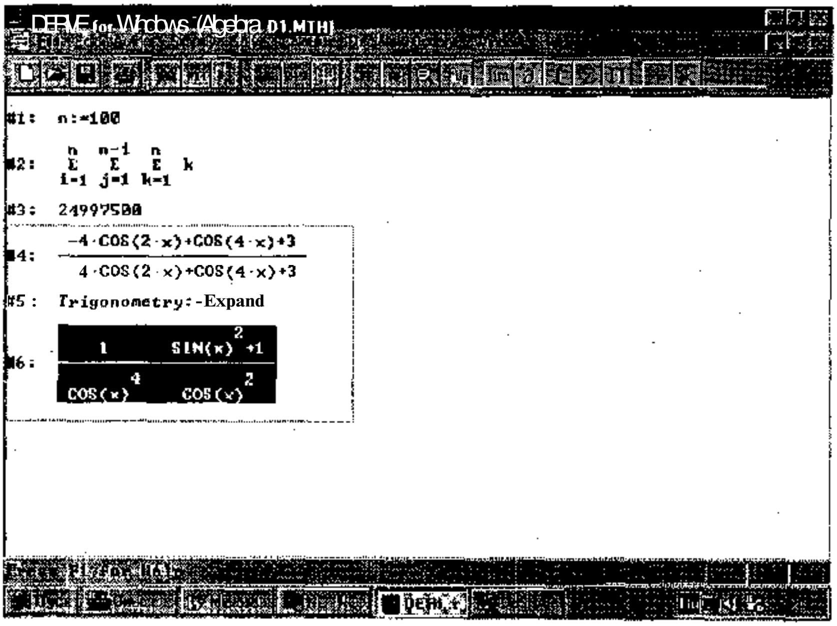


Рис. 6.8 Выделение с помощью команды Mark and Copy

Команда **Mark and Copy** позволяет выделить любой фрагмент в окне выражений или графиков и поместить его в буфер обмена. Для этого надо выполнить команду — указатель мыши приобретает вид тонкого креста — и, удерживая нажатой левую кнопку мыши, переместить прямоугольник выделения из пунктирных красных линий (рис. 6.8).

Примечание

Если какое-то из выделяемых выражений выделено еще и цветом, то это выделение сохраняется в графической копии выделенного фрагмента, находящейся в буфере обмена.

Ввод данных

Ввод выражений

В меню **Author** представлены команды, необходимые для ввода данных:

Expression — ввод математических выражений;

Vector — задание вектора указанных размеров;

Matrix — задание матрицы указанных размеров.

Команда **Expression** служит для ввода выражений, что уже описывалось (см. рис. 6.2).

Ввод констант и некоторых операторов

Для ускоренного ввода констант в Derive 4.* под Windows используются следующие сочетания клавиш:

Клавиши	Константа		Описание
Ctrl+Pp	π	pi	площадь единичного круга (3.14159...)
Ctrl+E	ё	#e	основание натурального логарифма (2.71828...)
Ctrl+I	î	#i	мнимая единица (квадратный корень из -1)
Ctrl+O	°	deg	число радиан в градусе **(p/180)**
Ctrl+0	00	inf	положительная бесконечность
Ctrl+U	∪	union	оператор объединения множеств
Ctrl+N	∩	intersection	оператор пересечения множеств
Ctrl+T	`	`	оператор транспонирования матриц/оператор дополнения множеств
Ctrl+B	↓	sub	оператор индекса векторов и матриц
Ctrl+Q	V	sqrt	квадратный корень

Ввод векторов

Для задания вектора используется команда Vector, которая вызывает появление окна для ввода единственного параметра — размера вектора. После ввода размера вектора появляется окно для ввода значений элементов вектора (рис. 6.9), которых может быть от 1 до 100. Если размер вектора достаточно велик, то справа от полей ввода появляется вертикальная полоса прокрутки.

Элементы векторов могут быть числами, именами переменных или символьными выражениями. На рис. 6.10 приведены примеры векторных операций.

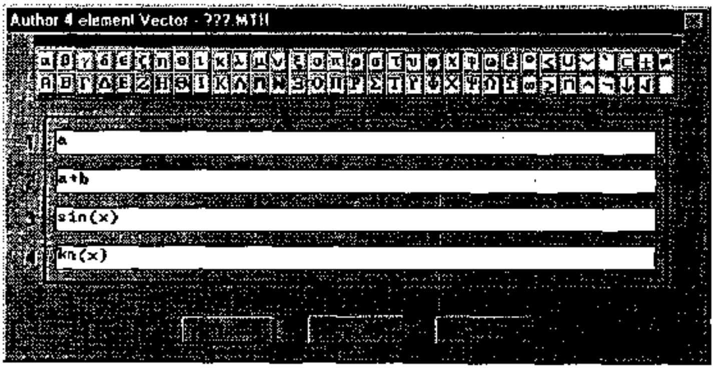


Рис. 6.9. Ввод элементов вектора

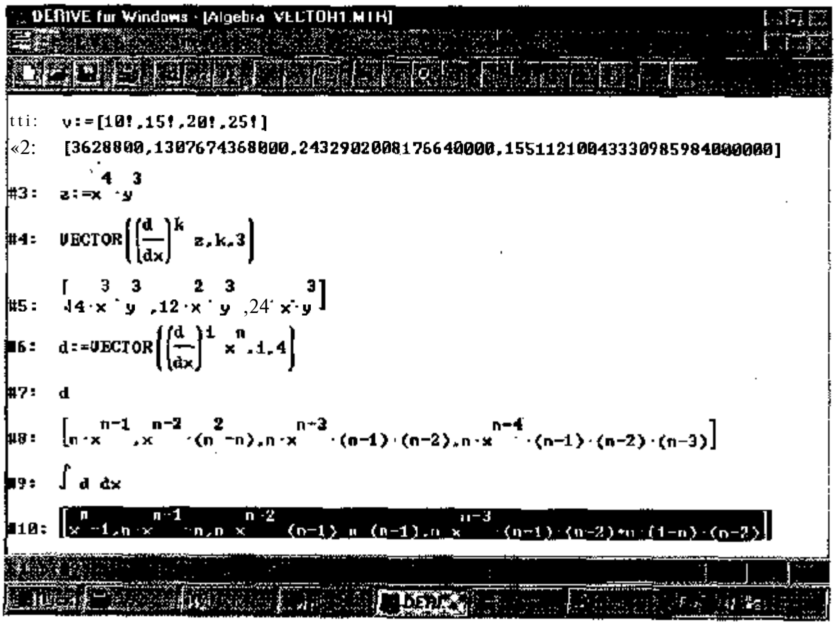


Рис. 6.10. Примеры выражений с векторами

Операторы доступа к элементам вектора ↓ и SUB

Для доступа к элементам вектора введен оператор ↓ (или **SUB**). Он используется следующим образом. Зададим вектор V := [1, 2, 3, 4]. Теперь зададим выражение **V↓2** или V **SUB** 2 и нажмем клавишу **Enter.** Получим

V_2

Исполнив команду **Simplify,** получим

2

Аналогично для выражения **V↓4** или V **SUB** 4 получим

V_4

4

Обратите внимание на то, что элементы вектора представляются как индексированные переменные, причем индекс записывается как подстрочный индекс (как это и принято в математике). К сожалению, операторы ↓ и **SUB** позволяют только извлечь содержимое заданного элемента вектора, но не задать ему новое значение. Функцию **VECTOR** и другие функции, используемые при работе с векторами, мы рассмотрим отдельно позже. Среди них, естественно, есть и средства для изменения значения заданного элемента вектора.

Ввод матриц

Матрицей называют двухмерную таблицу из m строк и n столбцов. Эти целые положительные числа определяют размер матрицы m×n. Для задания матрицы надо выполнить команду **Matrix** и в появившемся окне ввести значения m и n (ОТ 1 до 100) в поля **Rows** (строки) и **Columns** (столбцы). После ввода размера матрицы появляется окно для ввода элементов матрицы (рис. 6.11). Заметим, что матрица с размером 1x1 является единичным эле-

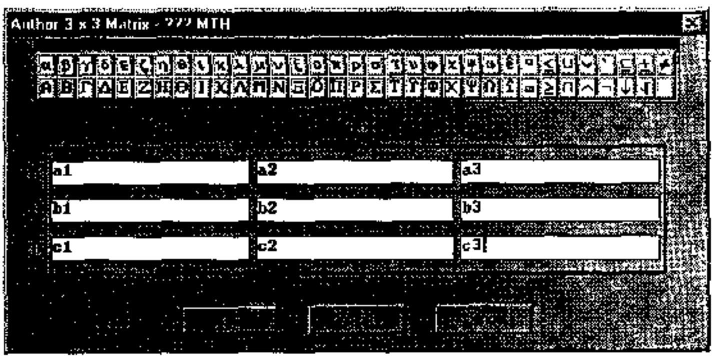


Рис. 6.11. Окно для ввода элементов матрицы

ментом, а матрица с одной строкой (m=1) — вектором. Вектор может располагаться и вертикально, если m>1, а n=1.

Если размер матрицы достаточно велик, в окне ввода появляются вертикальные и горизонтальные полосы прокрутки. Элементы матриц могут быть любыми числами, именами переменных и символьными выражениями (формулами).

Доступ к элементам и столбцам матриц (операторы ↓ и ↓↓)

Для доступа к элементам матриц можно использовать оператор **M↓m↓n** (или M SUB m SUB n), где m и n — номер строки и столбца матрицы M. Например:

$$M := \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Исполнив **M↓1↓1** (или **M** SUB **1** SUB 1), а затем Simplify, получим

$$M_{1,1}$$

1

Аналогично исполнив **M↓2↓2** (или M SUB 2 SUB 2), а затем Simplify, получим

$$M_{2,2}$$

4

Оператор ↓↓ в виде **M↓↓n** (или M SUB SUB n) возвращает вектор, содержащий n-й столбец матрицы M:

M↓↓1
[1, 3]
M↓↓2
[2,4]

Как и в случае векторов, операторы ↓ и SUB позволяют только извлекать содержимое заданного элемента матрицы.

Примеры работы с матрицами

На рис. 6.12 показаны примеры работы с матрицами, иллюстрирующие символьные вычисления, а также примеры на вычисление характеристического полинома и собственных значений матриц.

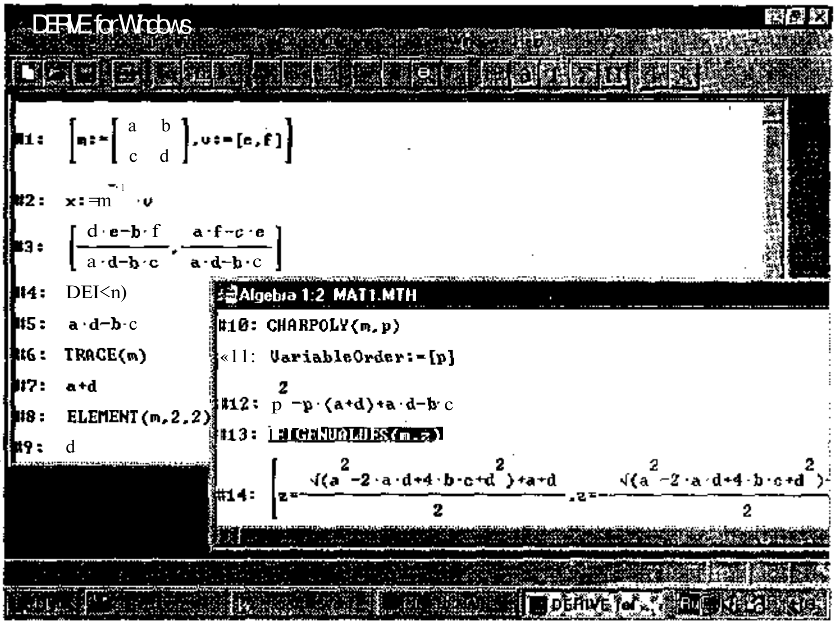


Рис. 6.12. Примеры работы с матрицами в системе Derive

Полезно обратить внимание, что векторы и матрицы в Derive являются списками и возможно создание списков в списках. В систему Derive включено множество векторных и матричных функций, с которыми мы ознакомимся позже.

Преобразование выражений

Важным этапом работы с системой является вычисление и преобразование введенных выражений и данных. Команды меню **Simplify** перечислены ниже:

Basic — общее упрощение выражений различного типа;

Expand — расширение полиномиальных выражений;

Factor — факторизация полиномиальных выражений;

Approximate — вычисление выражений в численном виде;

Substitute for — подстановка для переменной.

Назначение этому меню имени **Simplify** (упростить) выглядит не слишком удачным, поскольку под таким именем в системах символьной математики подразумевается одна функция упрощения, а не совокупность команд по преобразованию и вычислению выражений, как в Derive.

Упрощение выражений

Команда **Basic** применяется к выделенному выражению или части выражения. Она вызывает появление диалогового окна, подобного окну **Author Expression** (см. рис. 6.2), в котором имеется поле с выделенным выражением

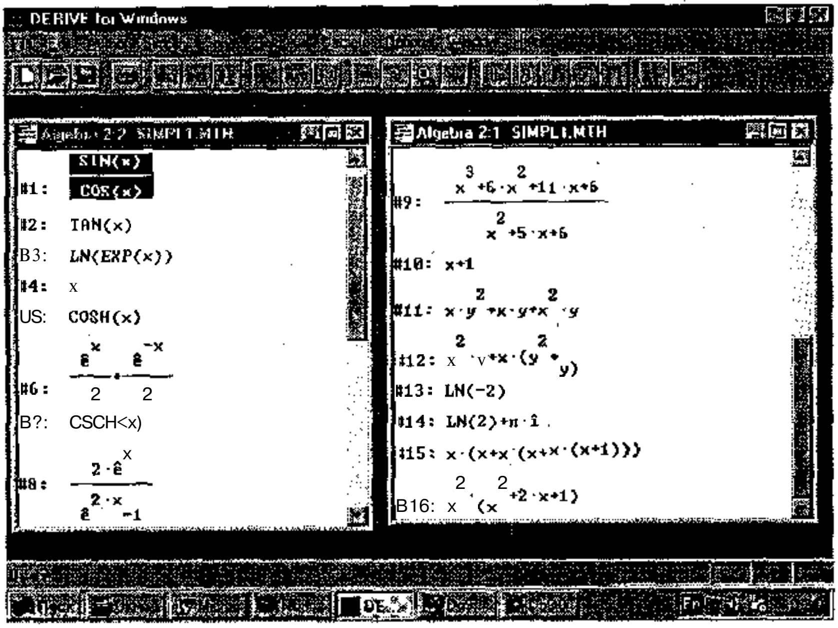
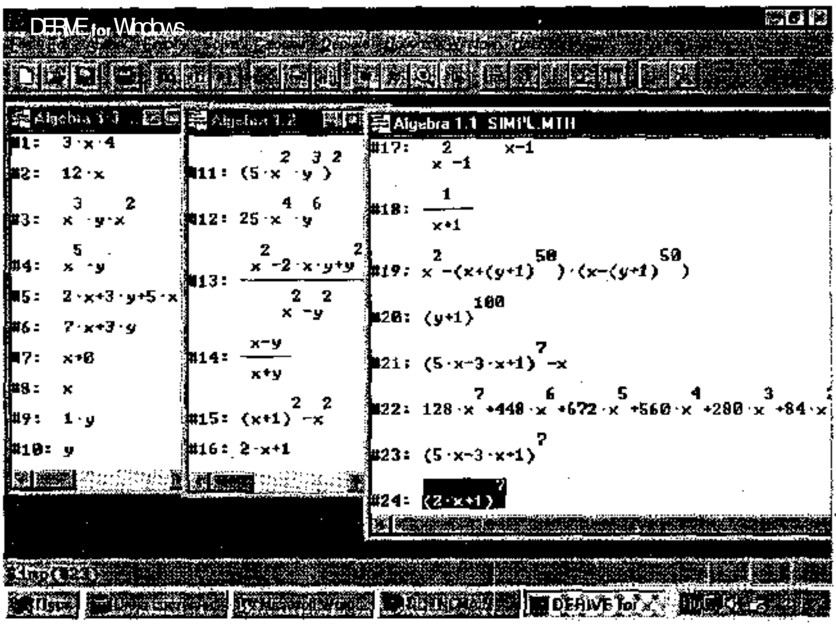


Рис. 6.13. Типовые примеры применения команды Basic



Рис; 6.14. Дополнительные примеры применения команды Basic и панель специальных символов. С их помощью можно корректировать выражение, для которого используется команда **Simplify • Basic.** На рис. 6.13 показан ряд типовых применений команды для алгебраических выражений (в основном полиномиальных). Для каждого примера приводится строка с исходным выражением и строка, полученная при применении команды.

Особый интерес представляет упрощение отдельных частей выражения, которое приводит подчас к совсем иным результатам, чем упрощение выражения в целом. Так, на рис. 6.14 в строках 21 и 22 упрощение выражения приводит к довольно длинному обычному полиному. Но если выделить выражение в скобках седьмой степени и упростить его, то окажется, что исходное выражение упростится к виду:

$$(2x+1)^7 - x$$

Это еще раз говорит о том, что упрощение выражений процесс творческий, и его результат зависит от подхода к упрощению.

Из примеров видно, что команда **Simplify • Basic** может применяться для упрощения тригонометрических и алгебраических выражений, содержащих переменные в степени и логарифмы.

Внимание

Обратите внимание на обозначение константы — основания натурального логарифма (см. рис. 6.13). Она обозначается буквой ё с птичкой наверху, что позволяет сразу отличать ее от обычной переменной е.

Расширение выражений

Расширение выражений с помощью команды **Expand** обычно используется для раскрытия полиномов и максимизации членов алгебраических выражений. Для рациональных выражений оно обычно дает разложение на элементарные дроби.

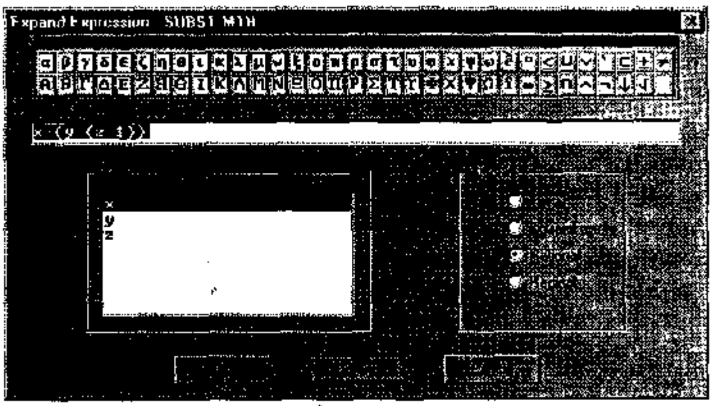


Рис. 6.15. Диалоговое окно команды Expand

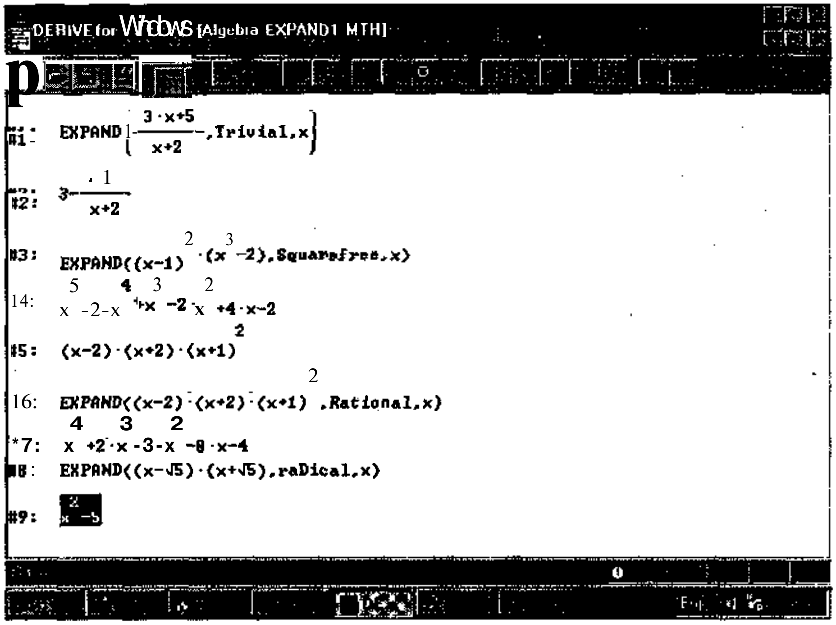


Рис. 6.16. Примеры применения команды **Expand**

В окне команды **Expand** (рис. 6.15) имеется список переменных, которые входят в расширяемое выражение. С помощью этого списка можно задать различные виды расширения относительно той или иной переменной. Кроме того, имеются переключатели для задания параметров расширяемого выражения: **Trivial** (тривиальное), **SquareFree** (свободное от радикалов), **Racional** (рациональное) и **Radical** (радикальное). На рис. 6.16 показаны примеры применения команды **Expand.**

Результаты преобразований с помощью команды **Expand** зависят от пути преобразований и выбора параметра расширяемого выражения. Команде **Expand** соответствует функция **EXPAND.**

Факторизация выражений

Факторизацией называют разложение на множители выделенных выражений или подвыражений, относительно одной или ряда переменных.

В окне команды **Factor** (рис. 6.17) имеются переключатели для различного представления выходного выражения: **Trivial** (тривиальное), **SquareFree** (свободное от радикалов), **Racional** (рациональное), **Radical** (радикальное) и **Complex** (комплексное). На рис. 6.18 показаны примеры факторизации выражений, представленных полиномами и их отношением.

Факторизация совместно с командой расширения **Expand** дает эффективные методы преобразования полиномиальных выражений. Рекомендуется внимательно просмотреть примеры применения команд **Expand** и **Factor,** чтобы убедиться в единстве их противоположности.

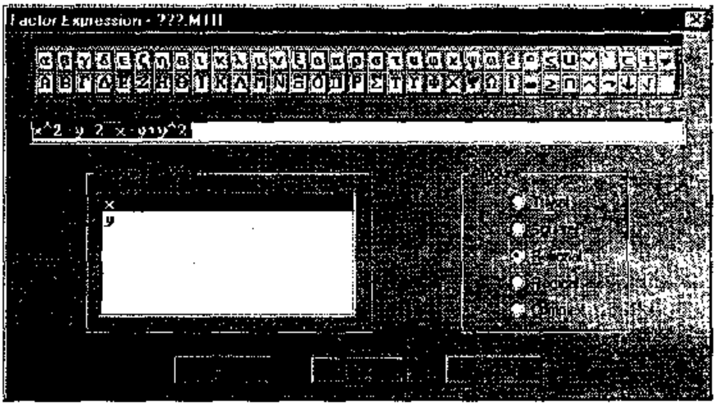


Рис. 6.17. Диалоговое окно команды Factor

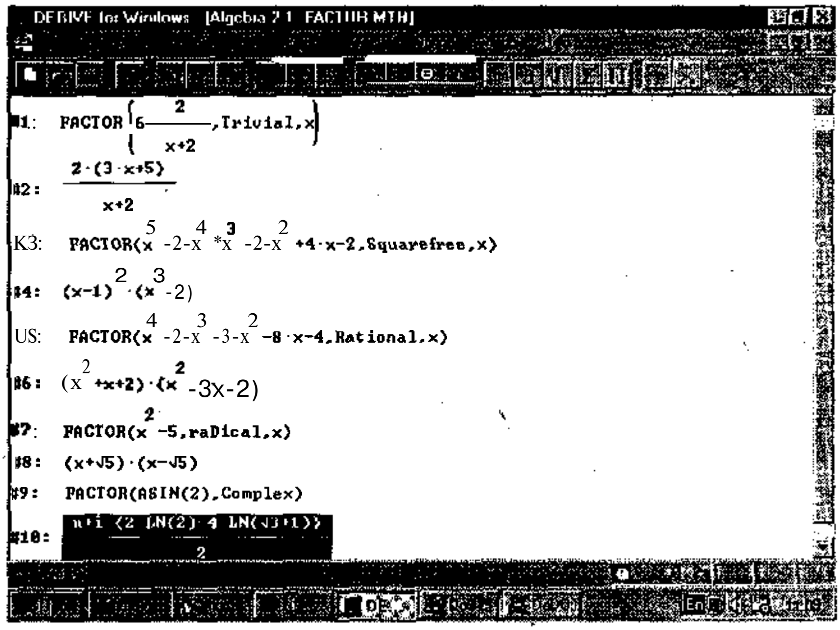


Рис. 6.18. Примеры факторизации для выражений с одной переменной

Вычисления выражений в численном виде

Для получения результатов в численном виде используется команда **Approximate,** окно которой показано на рис. 6.19.

Если в выделенном выражении переменные имеют численные значения, то команда **Approximate** использует их, пытаясь вычислить заданное выражение и представить результат вычисления в виде числа. На рис. 6.20 показаны примеры применения этой команды.

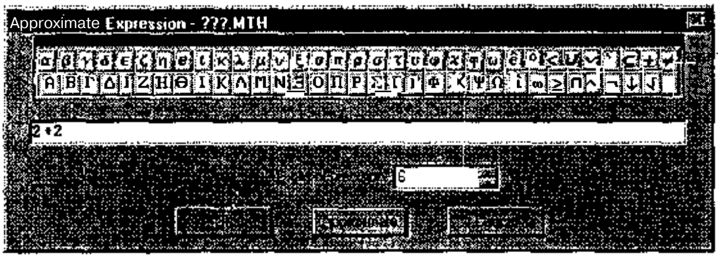


Рис. 6.19, Окно команды Approximate

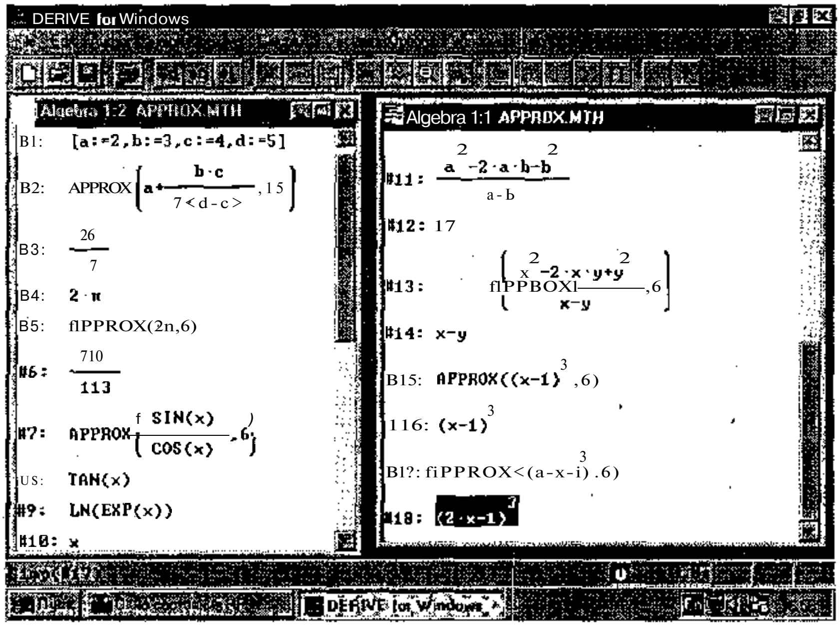


Рис. 6.20. Примеры применения команды Approximate

Если в выражении используются переменные, не имеющие численных значений, то команда **Approximate** (и функция **APPROX)** выполняет упрощение выражения.

Подстановка для переменной

Подстановки — еще одна важная возможность систем символьной математики. В Derive под Windows они реализованы двумя командами: **Substitute for Variable** — подстановка для переменной (рис. 6.21) и **Substitute for Subexpression** — подстановка для выделенного подвыражения.

Если выделено все выражение, то любой его переменной можно присвоить символьное или численное значение. Это открывает большие возможности в конструировании новых выражений для реализации тех или иных математических действий. На рис, 6.22 показано, как реализуется подстановка на

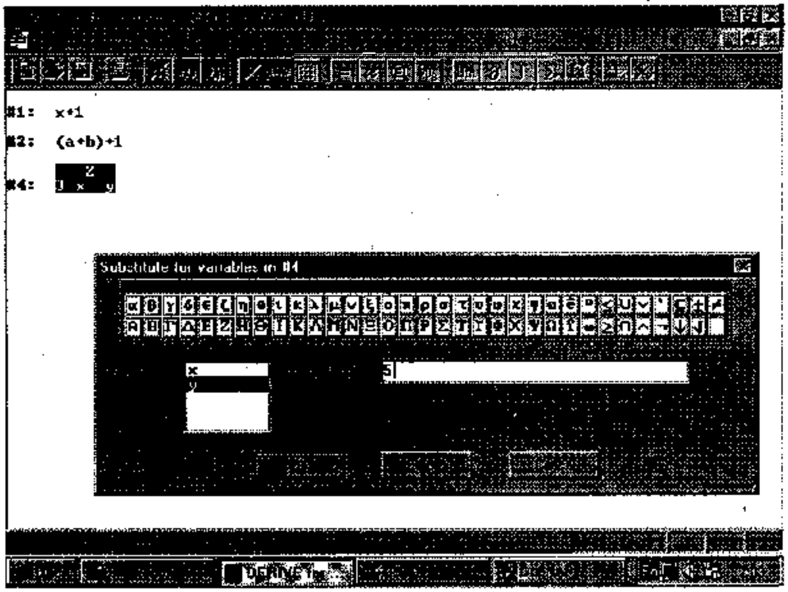


Рис. 6.21. Окно подстановки переменной

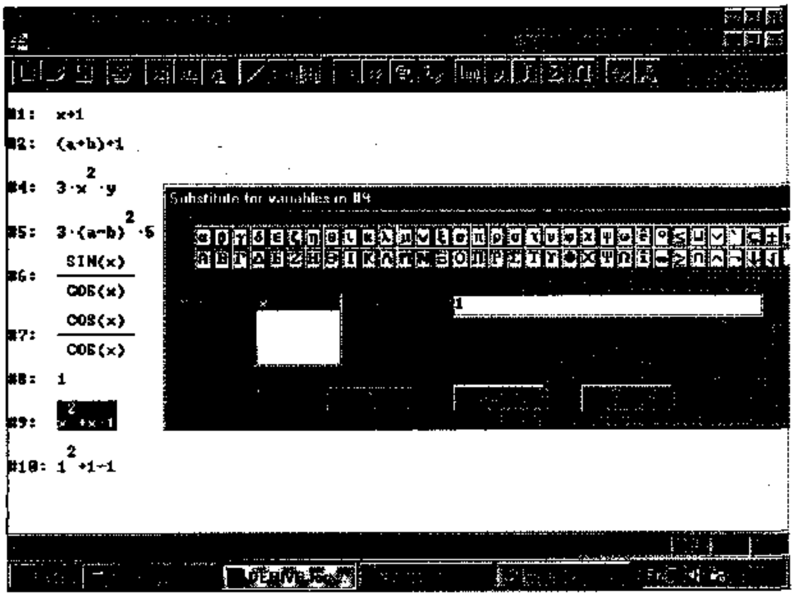


Рис. 6.22. Пример подстановки на место **переменной** ее численного значения

место переменной ее численного значения. Заметьте, что в этом случае подстановка производится без упрощения выражения, т. е. формально. Если полученное выражение надо вычислить, используйте команду Simplify • Basic или Simplify • Approximate.

Подстановки, особенно в символьной форме, излюбленный прием математиков. Они часто сводят сложные или даже невыполнимые преобразования к достаточно тривиальным.

Подстановка для подвыражения

Подстановка одного подвыражения на место другого выполняется командой Substitute for subexpression, окно которой показано на рис. 6.23. В окне в поле ввода нужно записать новое выражение. В нашем случае функция SIN(x) заменяется функцией COS(x).

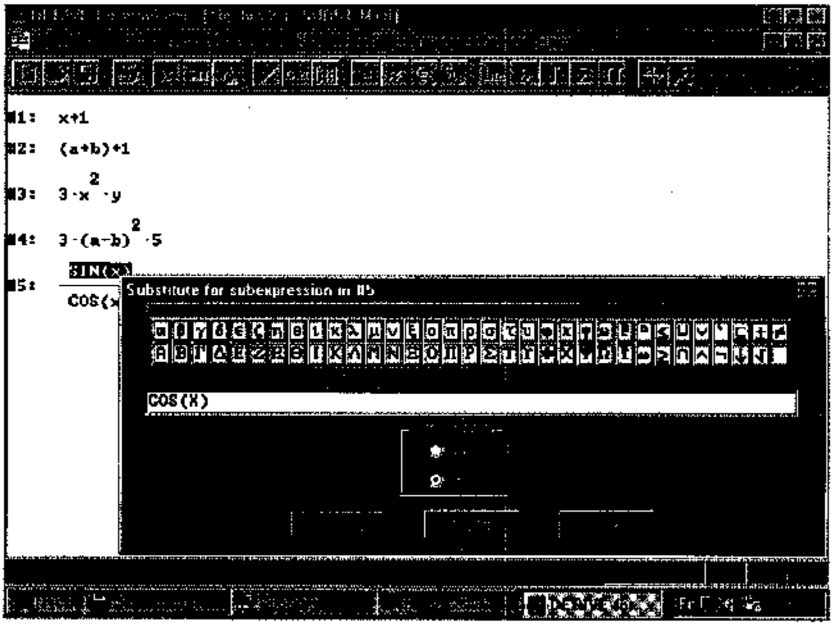


Рис. 6.23. Пример замены одного подвыражения другим

В окне этой команды в группе Occurrences имеются два переключателя: One — замена только выделенного подвыражения и **All** — замена всех подвыражений, совпадающих с выделенным.

Решение уравнений и неравенств

Для решения уравнений, неравенств и систем с ними служат команды меню Solve:

Algebraically — решение в символьном виде;

Numerically — решение в численном виде;

System — решение систем уравнений.

Рассмотрим особенности каждой из этих команд.

Решение в символьном виде

Для решения одиночных уравнений и неравенств в символьном виде используется команда Solve ▸ Algebraically (и функция SOLVE). В диалоговом окне этой команды имеется поле для редактирования выделенного уравнения или неравенства и раскрывающийся список для выбора переменной, относительно которой должно решаться уравнение или неравенство (рис. 6.24).

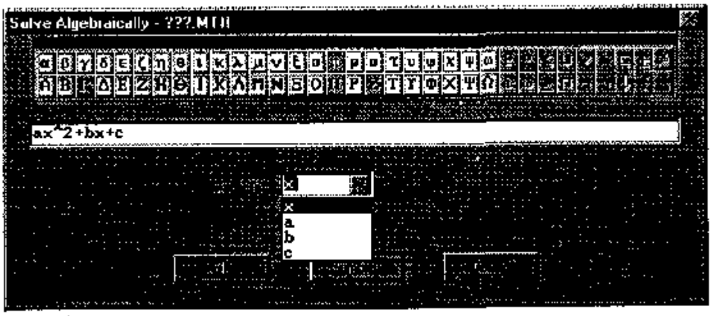


Рис. 6.24. Окно решения уравнений и неравенств в символьном виде

На рис. 6.25 представлено решение нескольких неравенств и квадратного уравнения с помощью команды Solve • Algebraically. В заключительном примере показано, что результатом решения может быть и отрицательная бесконечность.

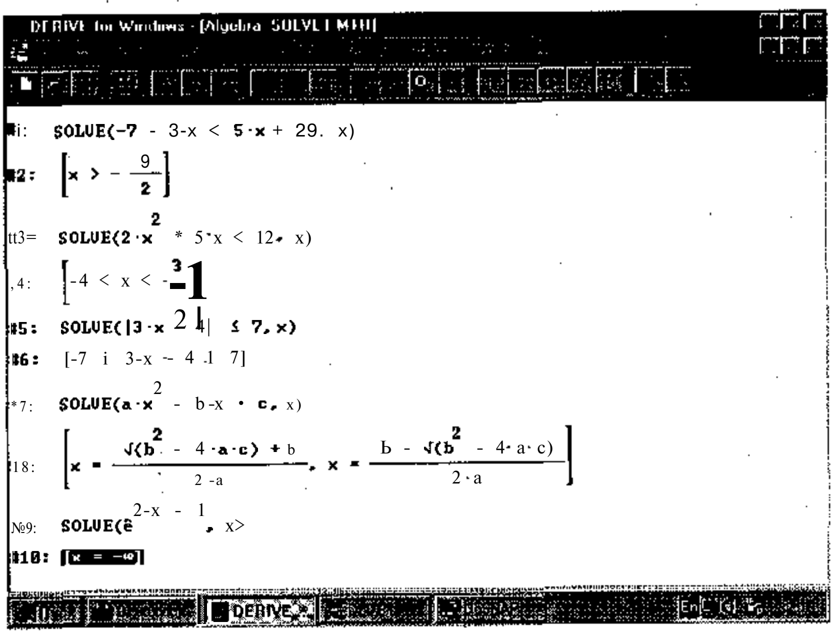


Рис. 6.25. Примеры решения неравенств и уравнений

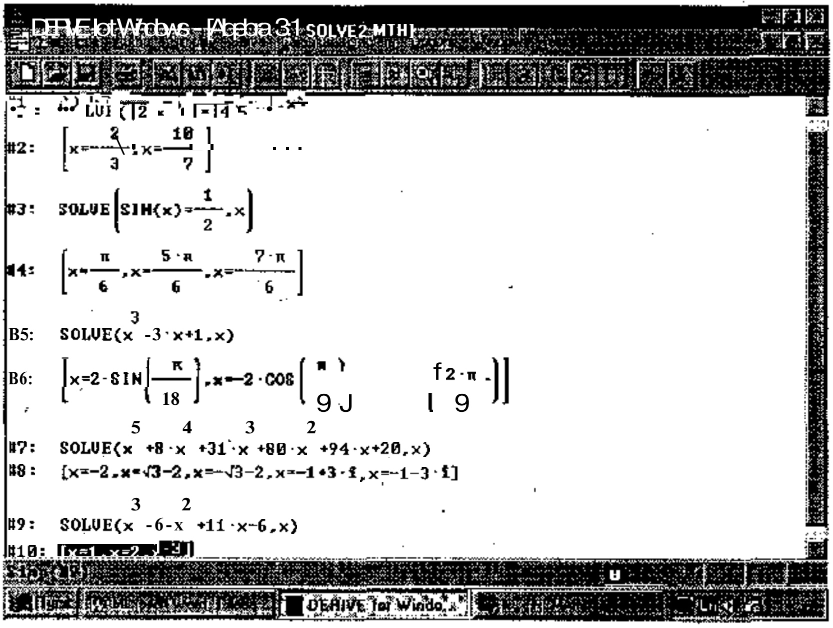


Рис. 6.26. Дополнительные примеры решения неравенств и уравнений

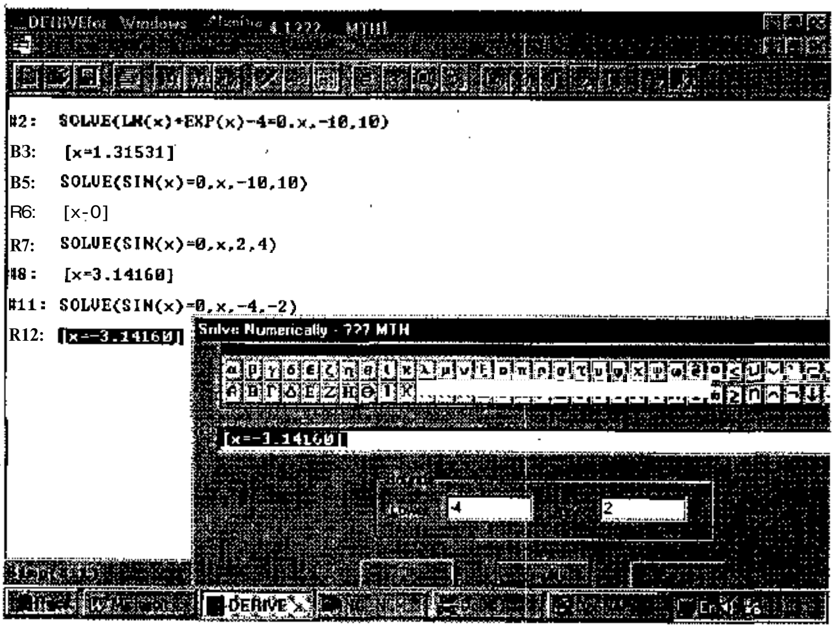


Рис. 6.27. Примеры решения уравнений в численном виде

Еще несколько примеров на применение команды **Solve • Algebraically** даны на рис. **6.26.** Здесь особенно полезно отметить два последних примера на поиск 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).

Решение в численном виде

Если необходимо решение уравнения в численном виде, то используется команда **Solve • Numerically.** Отличием этого вида решения от рассмотренного выше является не только представление результата в численном виде, но и задание в функции **SOLVE** пределов, в которых ищется корень (или корни) уравнения (рис. 6.27).

Данная команда порождает функцию **SOLVE(eqn,x,a,b)**. В функции **SOLVE** помимо самого уравнения eqn и переменной x, по которой оно решается, задается два предела a и b для области поиска корней. С другими формами записи этой важной функции можно познакомиться с помощью справочной системы Derive.

Решение систем уравнений

Если необходимо решать системы линейных уравнений относительно неизвестных, следует использовать команду **Solve • System.** При выборе этой команды вначале выводится окно с запросом на число уравнений в системе. После ввода числа уравнений появляется окно для ввода уравнений и указания переменных, относительно которых ищется решение (рис. 6.28). Для задания переменной надо щелкнуть мышью на ее имени в списке. Повторный щелчок снимает выделение.

После ввода уравнений и указания переменных формируется функция **SOLVE** для решения введенной системы. Система уравнений представляется в виде вектора, и переменные, относительно которых ищется решение, также представляются в виде вектора. Для решения системы достаточно выбрать

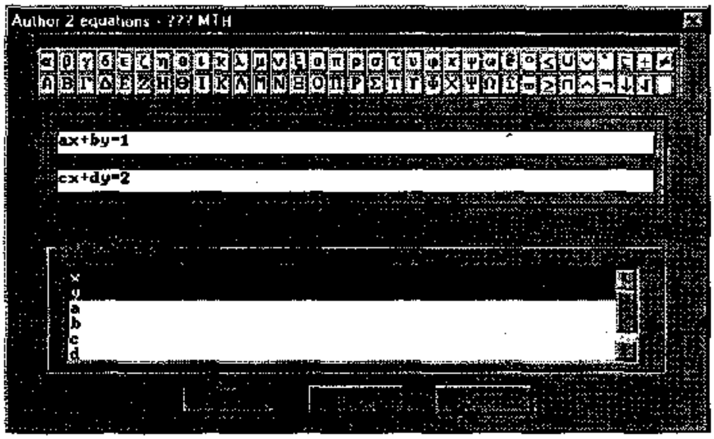


Рис. **6.28.** Окно ввода уравнений

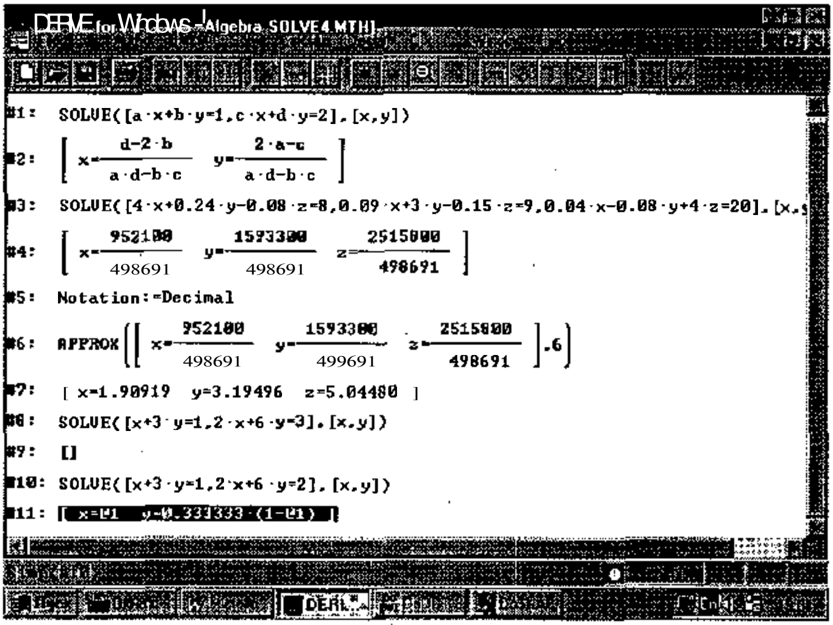


Рис. 6.29. Примеры решения систем уравнений

команду Simplify ▸ **Basic.** На рис. 6.29 приведен ряд примеров на решение систем линейных уравнений. Относительно других переменных уравнения могут быть нелинейными. Обратите внимание на два последних примера. В предпоследнем примере задана несовместная система уравнений. Поскольку в этом случае 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, то в решении появляются одно или несколько произвольных значений вида @n, где n — целое число. Количество таких значений во множестве решений равно общему числу уравнений за вычетом числа независимых уравнений.

Основные виды вычислений

В подменю Calculus сосредоточены наиболее важные команды для выполнения вычислений, относящихся к фундаментальным приложениям математического анализа:

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 эти команды более подробно.

Вычисление пределов функций

Для вычисления предела функции можно ввести ее как отдельное выражение и, выделив, выбрать команду Limit. Эту команду можно выполнить, не выделяя функцию, — ее придется ввести в поле ввода окна команды Limit, показанного на рис. 6.30.

Это окно позволяет также выбрать переменную, относительно которой ищется предел, и ввести ее значение (поле Limit Point). Кроме того, с помощью переключателей группы Approach From можно установить тип изменения переменной при поиске предела функции: подход слева (Left), справа

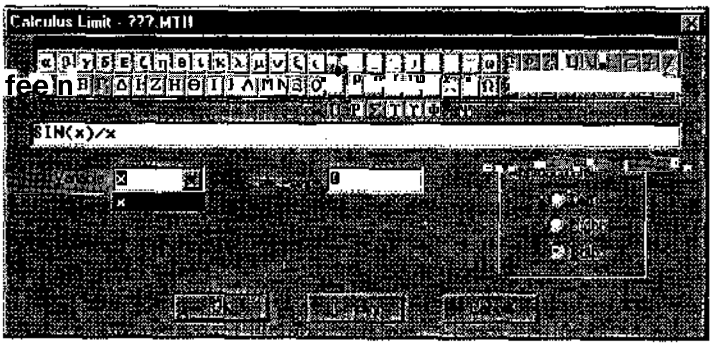


Рис. 6.30. Окно команды Limit

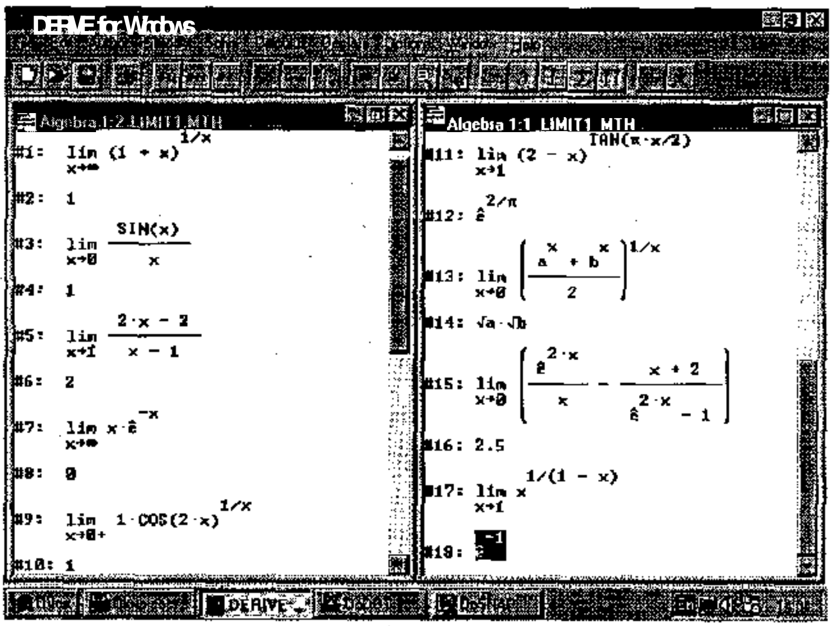


Рис. 6.31. Вычисление пределов различных функций

(Right) и с двух сторон (Both). Это необходимо, в частности, при поиске пределов разрывных функций.

Как обычно, окно команды **Limit** по завершении установок позволяет сразу вычислить предел (щелчком на кнопке **Simplify)** либо ввести функцию **LIMIT(u,x,a,[1 |-1]),** для вычисления которой необходимо выполнение команды **Simplify • Basic** (или **Simplify • Approximate** в случае нахождения предела в численной форме). На рис. 6.31 показан ряд примеров на вычисление пределов различных функций.

Необязательный параметр 1 или -1 функции **LIMIT** задает поиск предела выше или ниже точки а.

Вычисление производных функций

Для вычисления производных функций используется команда **Differentiate.** Она выводит окно (рис. 6.32), в котором задаются дифференцируемая функция, переменная, по которой осуществляется дифференцирование, и порядок искомой производной.

Щелчок на кнопке **OK** порождает функцию дифференцирования **DIFF(u,x[,n]).** Она вычисляет n-ю производную выражения и по переменной х или первую производную, если п опущено. Можно также найти производную без этой функции, щелкнув на кнопке **Simplify.** На рис. 6.33 представлен ряд примеров на дифференцирование различных функций.

Обратите внимание, что нередко операцию дифференцирования легко проверить интегрированием полученного выражения.

Разложение функций в ряд Тейлора

Для разложения функций в ряд Тейлора достаточно выбрать команду **Taylor series,** окно которой представлено на рис. 6.34.

В этом окне задается разлагаемая в ряд функция, переменная (раскрывающийся список **Variable),** относительно которой выполняется разложение, значение этой переменной в точке разложения (поле **Expansion Point)** и максимальный порядок членов разложения (поле со счетчиком **Order).** Щелчок на кнопке **Simplify** сразу возвращает результат разложения, тогда как щелчок на кнопке OK порождает функцию **TAYLOR(u,x,a,n)** со всеми необходимыми для ее вычисления параметрами: выражением и, переменной разложения х, точкой а и числом членов п. На рис. 6.35 показан наглядный пример на разложение функции SIN(x) в ряд Тейлора с максимальным порядком членов разложения 3, 5 и 7 (и числом членов ряда 4, 6 и 8). Обратите внимание на то, что для этой функции члены ряда с четными степенями нулевые.

Этот пример — хорошая иллюстрация к технике визуализации вычислений, проводимых с помощью Derive.

Вычисление неопределенных интегралов

Для вычисления неопределенных (без указания пределов интегрирования) и определенных интегралов используется команда **Integrate.** На рис. 6.36 показано окно этой команды, в котором можно задать подынтегральную функ-

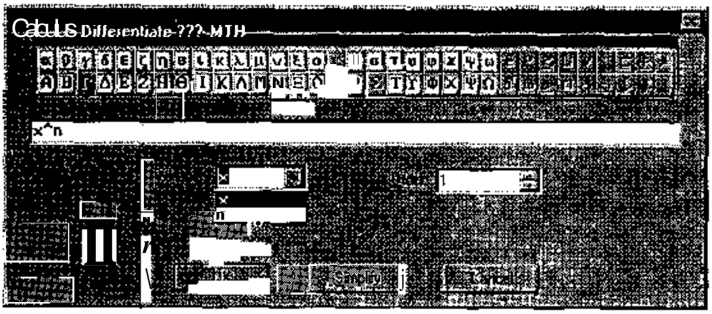


Рис. 6.32. Окно команды дифференцирования

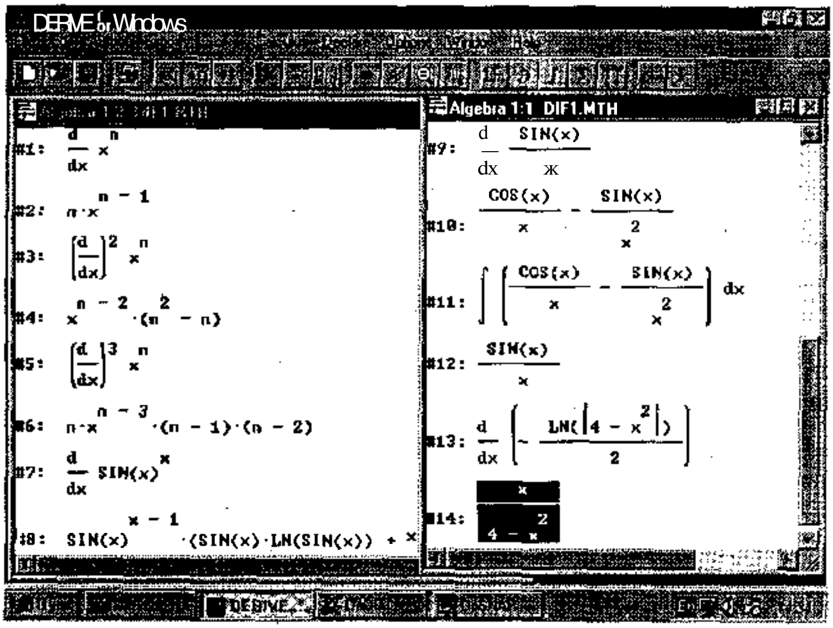


Рис. 6.33. Примеры дифференцирования различных функций

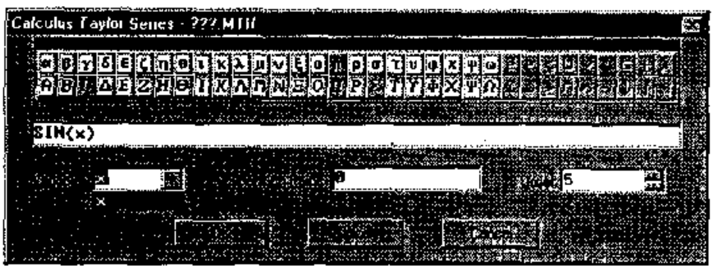


Рис. 6.34. Окно команды разложения функций в ряд Тейлора

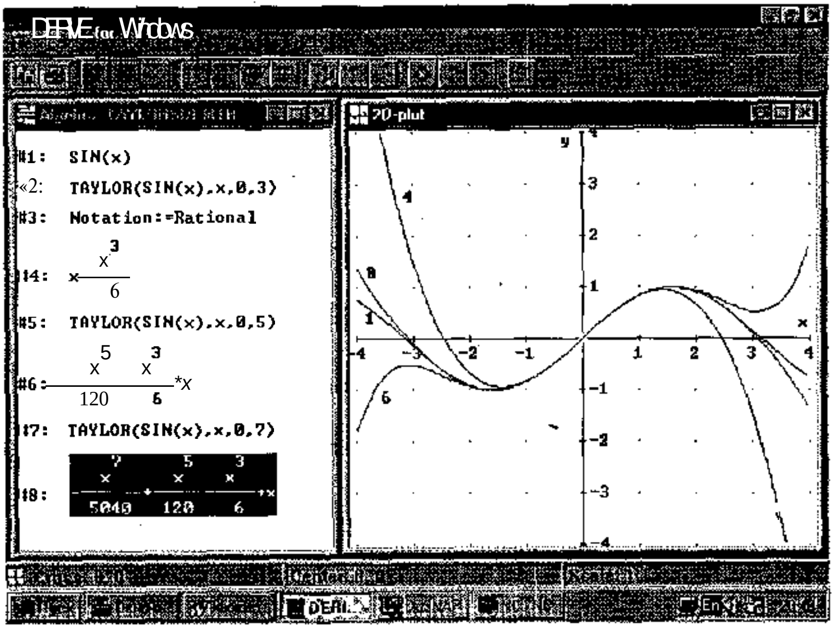


Рис. 6.35. Пример разложения синусоиды в ряд Тейлора

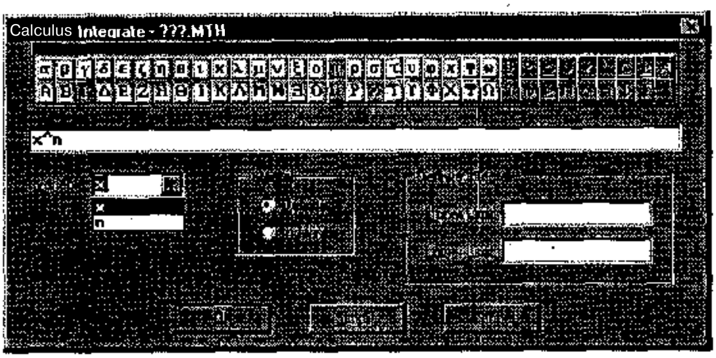


Рис. 6.36. Окно команды Integrate

цию, тип интеграла — определенный (переключатель **Definite)** или неопределенный (переключатель **Indefinite),** верхний (поле **Uper Limit)** и нижний (поле **Low Limit)** пределы интегрирования (для определенных интегралов), а также переменную, по которой осуществляется интегрирование.

Щелчок на кнопке **OK** порождает функцию **INT(u,x[,a,b])** со всеми нужными для ее вычисления параметрами. Кнопка **Simplify** позволяет вычислить интеграл сразу и возвращает *его* значение (символьное или численное). Для вычисления первообразных служит функция **DIF(u,x,–n)**, в которой п указывается со знаком минус. На рис. 6.37 показано вычисление однократных простых интегралов с полиномиальными и рациональными подынтегральными функциями.

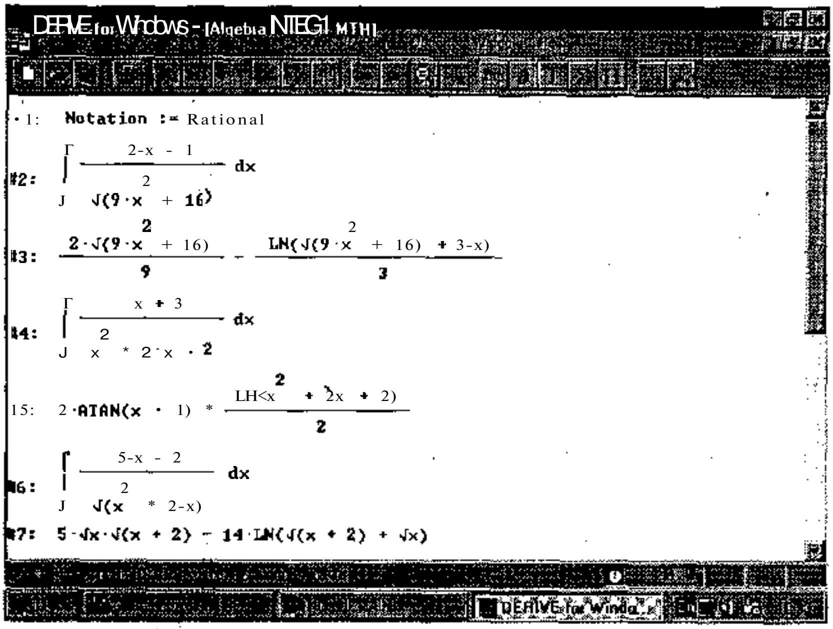


Рис. 6.37. Вычисление неопределенных интегралов с рациональными подынтегральными функциями

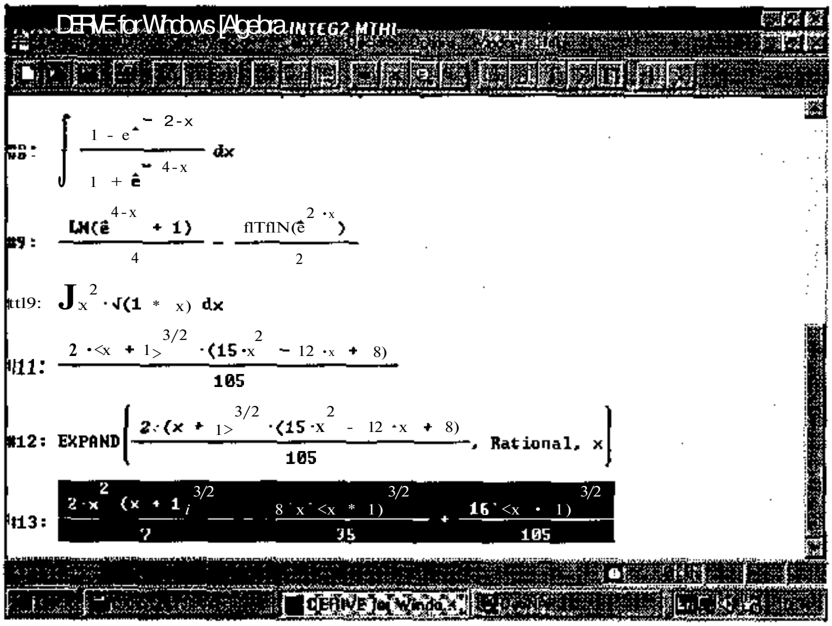


Рис. 6.38. Вычисление неопределенных интегралов с алгебраическими выражениями

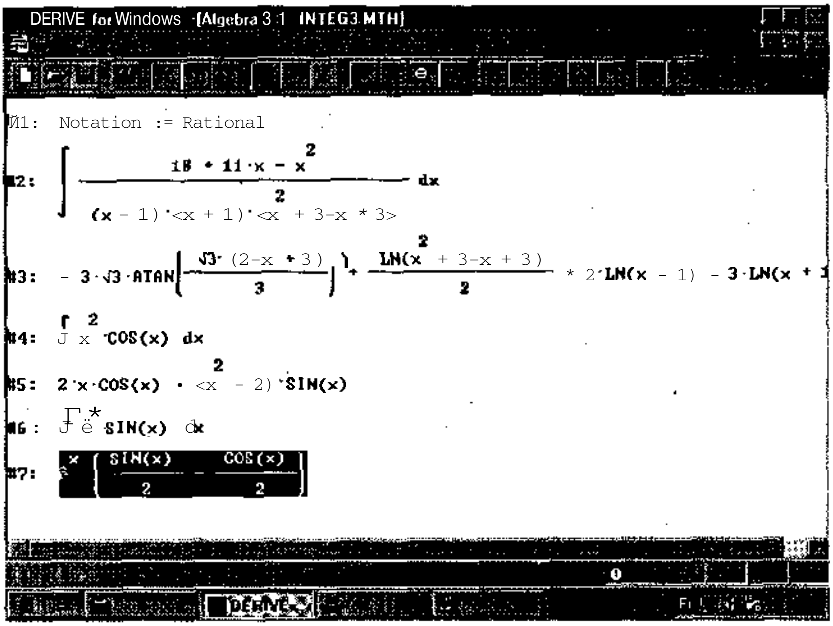


Рис. 6.39. Дополнительные примеры на вычисление неопределенных интегралов

В примерах на вычисление интегралов, показанных на рис. 6.38, подынтегральные функции — алгебраические выражения, содержащие логарифмы и экспоненциальные функции.

Дополнительные примеры на вычисление неопределенных интегралов даны на рис. 6.39.

Вычисление определенных **интегралов**

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 они выглядят обманчиво просто) приведены на рис. 6.40.

И наконец, на рис. 6.41 представлен ряд примеров вычисления кратных (двойных и тройных) интегралов, как неопределенных, так и определенных.

Derive неплохо справляется с вычислением интегралов, у которых подынтегральная функция и результат представимы выражениями с элементарными функциями. Для вычисления интегралов, дающих представление в виде специальных математических функций, система Derive приспособлена слабо, что, однако, может компенсироваться библиотеками расширения Derive.

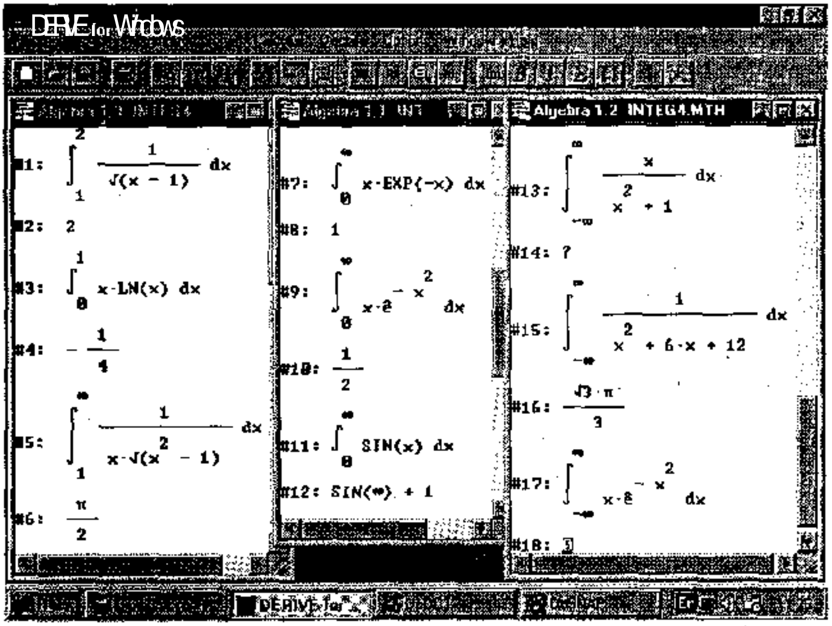


Рис. 6.40. Примеры вычисления несобственных интегралов

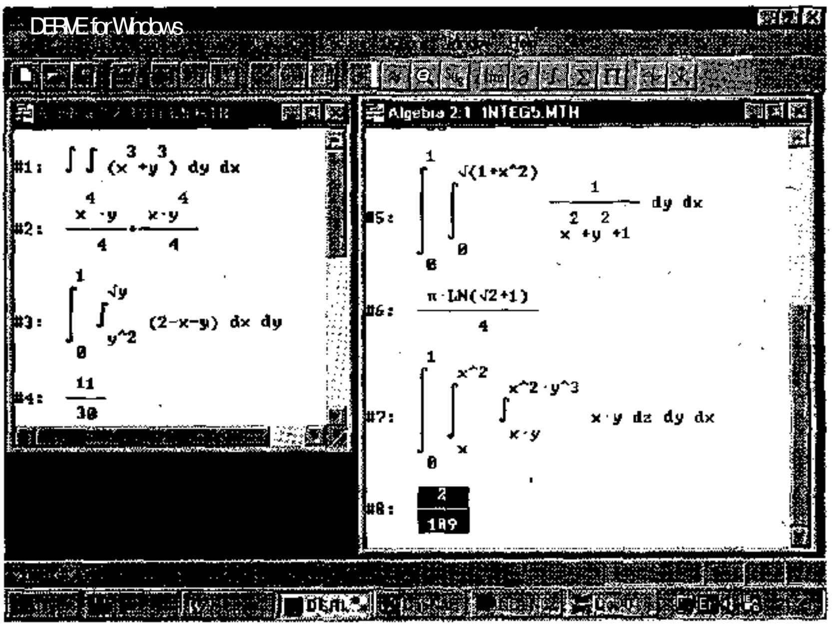


Рис. 6.41. Вычисление кратных интегралов

Вычисление суммы членов ряда

Для вычисления суммы членов ряда используется команда **Sum,** окно которой показано на рис. 6.42. В нем задается функция, значения которой суммируются, индексная переменная и верхний и нижний пределы ее изменения. Кроме того, можно задать определенную (переключатель **Definite)** и неопределенную (переключатель **Indefinite)** сумму. В последнем случае вычисляется ряд с бесконечным числом членов.

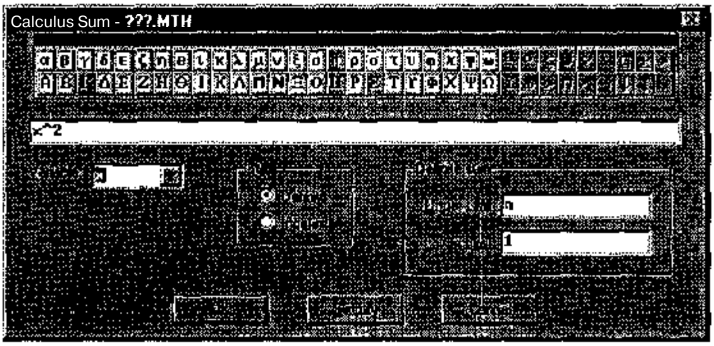


Рис. 6.42. Окно вычисления суммы членов ряда

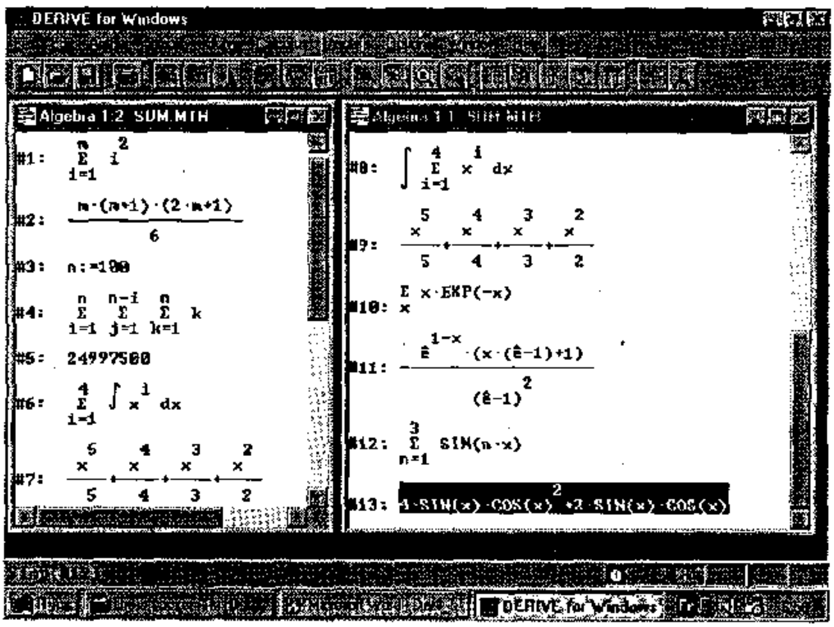


Рис. 6.43. Примеры вычисления суммы членов ряда

На рис. 6.43 представлено большое число примеров на вычисление суммы членов ряда. Нетрудно заметить, что система Derive способна вычислять как символьные, 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.

Сумма членов выражения и вычисляется с помощью функции **SUM(u,n[,k,m])** — в общем случае при изменении п от к до т.

Вычисление произведений членов ряда

Вычисление произведений членов ряда выполняется с помощью команды **Product.** Окно этой команды подобно окну вычисления сумм. В нем задается функция **PRODUCT(u,n[,k,m])** для вычисления произведений членов ряда при изменении п от к до т. Можно также задать произведение с определенным числом членов и неопределенным — индексная переменная п в последнем случае меняется до бесконечного значения. На рис. 6.44 представлен ряд примеров вычисления произведения членов ряда. Здесь также полезно обратить внимание, что вычисляются произведения как в аналитическом, так и численном виде.

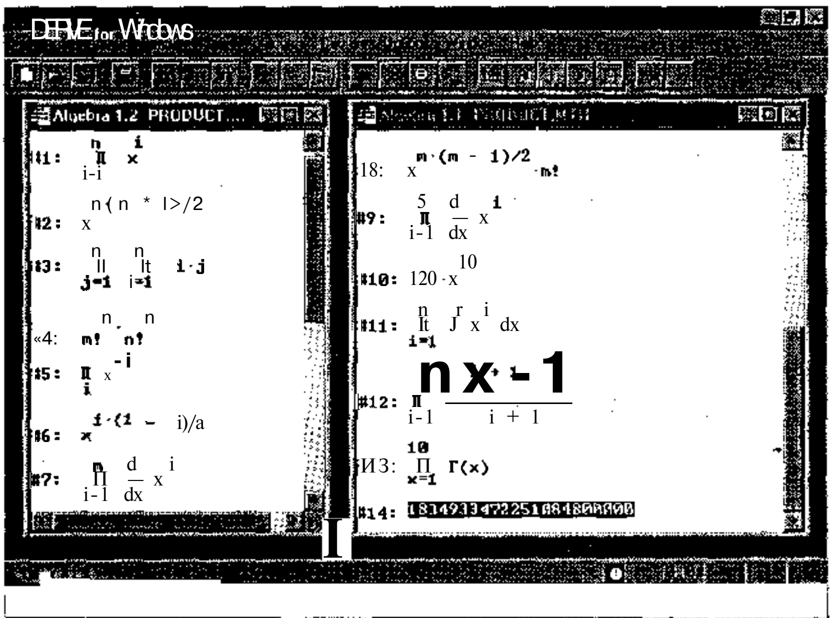


Рис. 6.44. Примеры вычисления произведений членов рядов

Вычисления сумм и произведений рядов широко используются в специальных математических функциях, представленных через суммы и произведения.

Представление ряда в виде вектора

Функция **VECTOR(u,k,m,n,s)** может генерироваться с помощью специальной команды — **Vector,** окно которой показано на рис. 6.45.

Помимо функции и индексной переменной, в этом окне задаются: начальное значение индексной переменной k(m) (поле **Starting Value),** конечное значение индексной переменной k(n) (поле **Ending Value)** и шаг изменения индексной переменной **k(s)** (поле **Step Size).** Примеры применения функции **VECTOR** неоднократно приводились выше. Допускается также применение данной функции для реализации циклов. Возможны упрощенные формы записи, например, **VECTOR(u,k,n)** соответствует изменению **к** от 1 до п с шагом + 1.

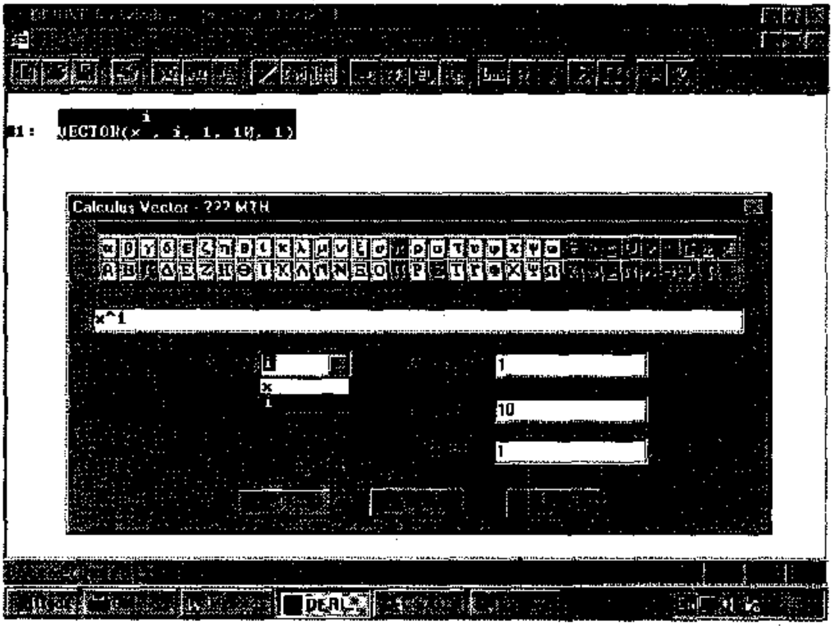


Рис. 6.45. ОКНО задания функции VECTOR

Объявление переменных и функций

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 конкретные значения. Кроме того, поскольку набор встроенных в ядро Derive функций ограничен, то часто возникает необходимость задавать функции одной или ряда переменных. Все эти возможности (а также задание формата ввода и вывода) дают команды меню **Declare:**

Variable Value — объявление переменной и присвоение ей значения;

Variable Domain — объявление области определения переменной;

Function Definition — задание функции пользователя;
Algebra State — вызов подменю управления форматом ввода/вывода.
Рассмотрим их более подробно.

Объявление переменной и присвоение ей значения

Для объявления переменной и присвоения ей численного или символьного значения используется команда **Variable Value,** в окне которой можно задать имя переменной и ее значение.

Объявление области определения переменной

Derive позволяет использовать обширный класс значений или областей определения переменных. Переменные могут быть с целочисленными, рациональными, комплексными и иными значениями. Для объявления области определения некоторой переменной служит команда **Variable Domain.** Выбор этой команды ведет к открытию окна с полем задания имени переменной. После задания переменной выводится окно для выбора типа переменной и области ее определения (рис. 6.46).

В левом верхнем углу окна выводится имя переменной, для которой объявляется область определения. Тип переменной можно выбрать с помощью переключателей группы **Domain,** а область определения — с помощью переключателей группы **Interval.**

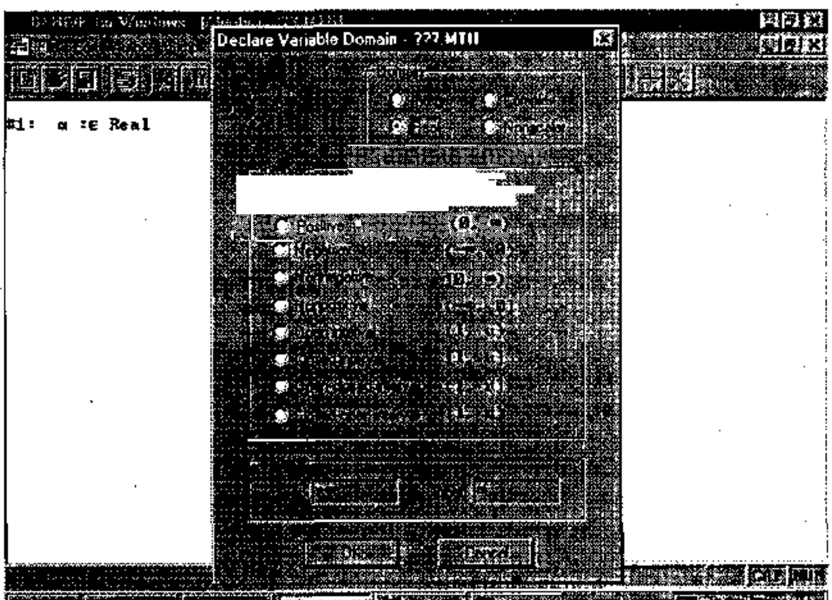


Рис. 6.46. Окно объявления области определения заданной переменной

Задание функции пользователя

Функции пользователя в системе Derive под Windows задаются командой **Function Definition,** окно которой показано на рис. 6.47.

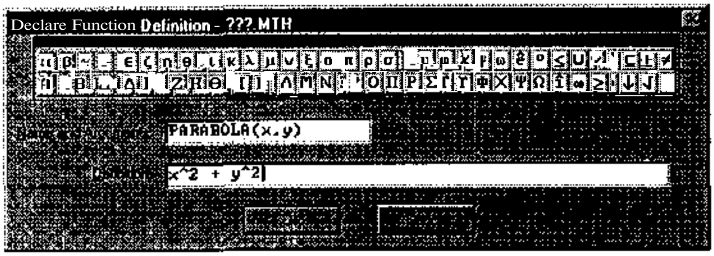


Рис. 6.47. Окно задания функции пользователя

В этом окне достаточно задать необходимую функцию и щелкнуть на кнопке ОК. В результате заданная функция будет введена в очередную строку документа и станет доступной для применения. Функции пользователя имеют вид

имя(параметры):=тело функции

Например, CS(x,y):=SIN(x)+COS(y). Параметры задаются списком переменных, которые в теле функции (выражении) являются локальными. Функции пользователя могут быть рекурсивными, т. е. допускать в теле функции обращение к самой себе.

Подменю управления форматом ввода/вывода

Позиция **Algebra State** подменю **Declare** содержит подменю со следующими командами:

Input — выводит окно параметров ввода;

Output — выводит окно параметров вывода;

Simplification — выводит окно параметров упрощения;

Reset AH — отмена всех изменений.

Параметры, задаваемые с помощью этих команд, действуют глобально до отмены всех изменений командой **Reset All.**

Окно задания параметров ввода показано на рис. 6.48.

С помощью переключателей группы **Input Mode** можно задать режим ввода, т. е. воспринимать ряд следующих друг за другом символов как отдельные символы (переключатель **Character)** или как слово (переключатель **Word).** Это имеет значение при вводе идентификаторов (имен) переменных. В первом случае ввод последовательности символов xyz воспринимается как умножение x на y и на z, а во втором — как ввод переменной с именем xyz. С помощью переключателей группы **Case Sensitivity** можно задать чувствительность (переключатель **Sensitivity)** или нечувствительность (переключатель **Insensitivity)** к регистру вводимых символов. По умолчанию установлены переключатели **Character** и **Insensitivity.**

С помощью раскрывающегося списка **Radix** можно выбрать систему счисления: **Binary** (двоичная), **Octal** (восьмеричная), **Decimal** (десятичная) и **Hexadecimal** (шестнадцатеричная). По умолчанию выбрана десятичная система счисления. Назначение кнопок в нижней части окна следующее:

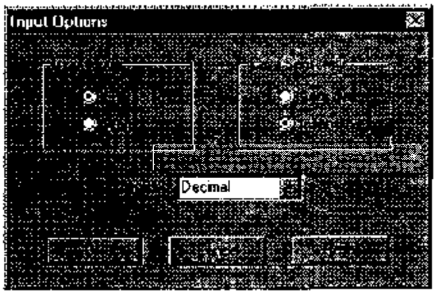


Рис. 6.48. Окно задания параметров ввода

OK — фиксация ввода;

Cancel — отмена внесенных изменений параметров ввода;

Reset — отмена внесенных изменений и возвращение к параметрам, принятым по умолчанию.

Окно задания параметров вывода показано на рис. 6.49.

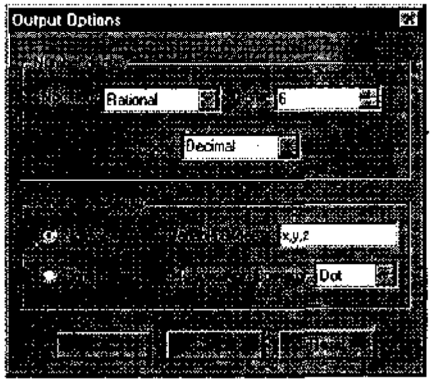


Рис. 6.49. Окно задания параметров вывода

Формат чисел можно выбрать в раскрывающемся списке **Notation** (десятичный — **Decimal,** смешанный — **Mixed,** рациональный — **Racional** и комплексный — **Complex),** число цифр числа — в поле со счетчиком **Digits** (по умолчанию 6) и систему счисления — в раскрывающемся списке **Radix** (он аналогичен списку для параметров ввода). Параметры представления выражений обеспечивают вывод в нормальной (переключатель **Normal)** и сжатой (переключатель **Compesed)** форме, а также установку порядка переменных (по умолчанию x, y, z) и знака операции умножения (символом * — Asterisk, точкой — Dot и пробелом — Implicit).

В окне параметров упрощения (рис. 6.50) можно выбрать направление преобразования (группа **Transformation Direction),** единицы представления углов (раскрывающийся список **Angular unit),** точность (группа **Precision)** и ветвь преобразований (раскрывающийся список **Branch).**

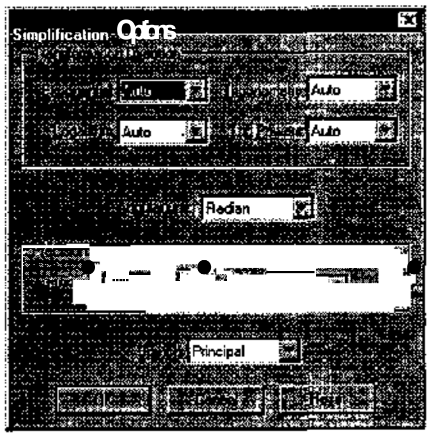


Рис. **6.50.** Окно задания параметров упрощения

Для экспоненциальных (Exponencial), тригонометрических (Trigonometry) и логарифмических (Logarithm) функций задается направление преобразования трех видов: **Auto** — автоматически, **Collect** — сбор и **Expand** — расширение. Для функций тригонометрических степеней (Trig Power) направление преобразования задается значениями **Auto, Cosines** (преобразование к косинусам) и **Sines** (преобразование к синусам).

Точность (Precision) задается режимом (раскрывающийся список **Mode)** с возможными значениями **Approximate, Exact** и **Mixed,** а также числом цифр (поле со счетчиком **Digits).** Для ветви преобразования возможны значения **Principial, Real** и **Any.**

Чему мы научились:

- Работать с Derive 4 под Windows.
- Работать с меню и интерфейсом пользователя.
- Работать с файлами.
- Редактировать выражения и документы.
- Вводить данные.
- Преобразовывать выражения.
- Решать уравнения и неравенства.
- Выполнять **основные** виды вычислений.
- Задавать переменные и функции.
- Управлять форматом ввода/вывода.

Урок 7. Управление системой Derive 4 под Windows

- Установка параметров системы
- Работа с окнами
- Графические средства
- Справочная система

Установка параметров системы

Некоторые параметры системы (например, цвет символов и фона окна, параметры печати и др.) носят глобальный характер и именуются опциями. Они задаются с помощью меню **Options,** в которое входит подменю и две команды — **Color** и **Printing.**

Подменю **Color** содержит команды для установки параметров цвета:

Windows Text — цвет текста;

Windows Background — цвет фона;

Highlight Text — цвет выделенного текста;

Highlight Background — цвет фона выделенного текста;

Команда **Printing** открывает диалоговое окно, в котором с помощью переключателей задаются следующие параметры печати:

Label — печать меток, т. е. номеров строк;

Communication time — печать времени вычислений;

Annotation — печать комментариев к выражениям.

Команда **The Entry Dialog** — включение/выключение режима диалога при вводе. Команда действует как флажок: диалог включается, если против имени команды установлена галочка, и выключается, если ее нет.

Примечание

Качественная и правильная печать выполняется специальным шрифтом DfW Printer, который устанавливается автоматически при установке самой системы Derive.

Работа с окнами

В подменю **Window** расположены основные команды управления окнами и элементами пользовательского интерфейса:

New Algebra View — создание дополнительного окна для выражений;

New 2D-plot Window — создание дополнительного окна для двухмерной графики;

New 3D-plot Window — создание дополнительного окна для трехмерной графики;

Cascade — каскадное расположение окон;
Tile Horizontally — расположение окон по горизонтали;
Tile Vertically — расположение окон по вертикали;
Arrange Icons — упорядочить расположение кнопок (значков) свернутых окон;
Toolbar — отображение или скрытие панели инструментов;
Status Bar — отображение или скрытие строки состояния.
Рассмотрим возможности задания и вывода окон.

Создание дополнительного окна для выражений

Иногда выражений в документе так много, что они не умещаются в одном окне. Можно открыть дополнительное окно для выражений, используя команду **New Algebra View.** В этом окне будут присутствовать те же выражения, что и в исходном (начальном) окне. Как новые окна, так и старые можно менять в размерах и перетаскивать.

Средства графики

Создание нового окна для двухмерной графики

Команда **New 2D-plot Window** создает новое окно для построения двухмерных графиков. Такое окно имеет собственные панель инструментов и кнопки свертывания, восстановления и закрытия окна. Имеет оно и свое главное меню, несколько отличное по набору позиций и команд от меню окна выражений.

Основными объектами окна двухмерной графики являются: строка заголовка, строка меню, панель инструментов, сам график, графический курсор (в виде креста +), координатная система, область выделения части графика (показана пунктирными линиями) и строка состояния, в которой отображаются координаты курсора (Cross), координаты точки пересечения осей (Center), которая находится в центре окна с графиком, и масштабные множители по горизонтальной и вертикальной оси (Scale).

Графический курсор может перемещаться клавишами управления курсором. Для увеличения шага 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.

Панель инструментов окна двухмерной графики

Панель инструментов окна двухмерной графики дает эффективные и быстрые средства построения и модификации графиков. При этом практически не требуется обращение к командам меню, хотя их набор несколько больше, чем набор команд, доступных с панели инструментов. Панель инструментов содержит шесть групп кнопок быстрого управления построением графиков.

Копирование и печать графика:

Copy plot window — копирование графика в буфер обмена;

Print — печать графика.

Управление построением графика:

Create annotation — создание комментария, располагаемого по месту 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 на 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 (F6) — 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.

Возвращение в окно выражений:

Algebra window **(Ctrl+1).**

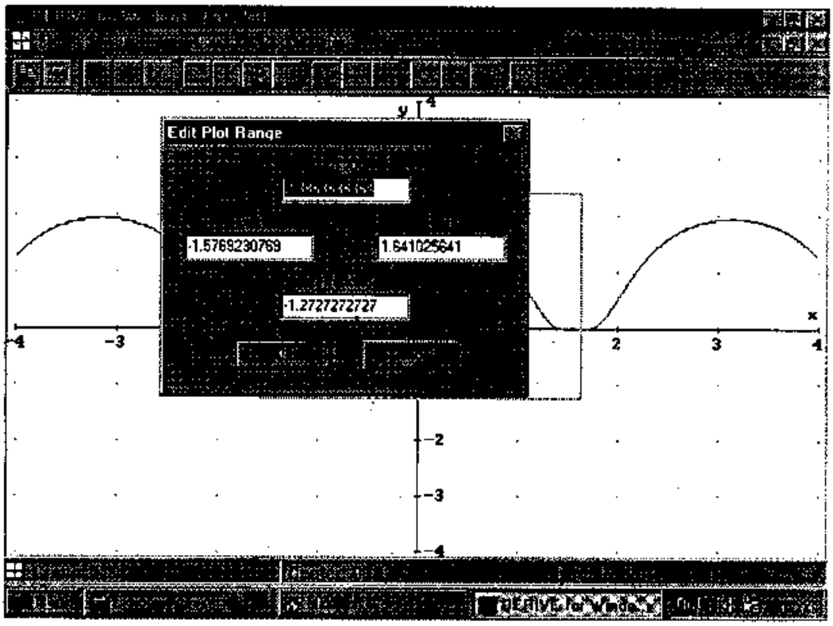


Рис. 7.1, Уточнение размера выделенного фрагмента рисунка

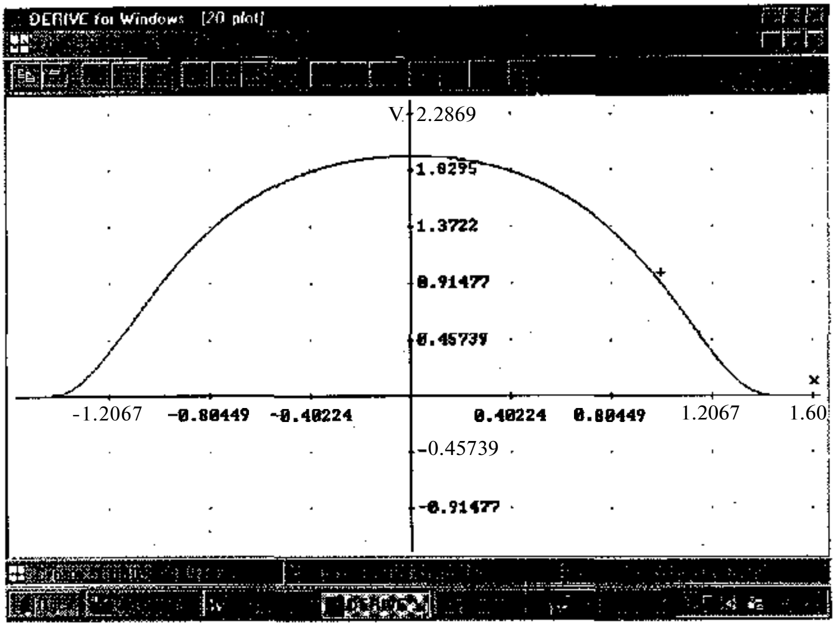


Рис. 7.2. График выделенного фрагмента

Чтобы понять, как работать с **панелью инструментов,** достаточно всего нескольких экспериментов. Поэтому остановимся лишь на одной, интересной возможности — выделении части изображения с помощью кнопки **Set range with box.** После щелчка на этой кнопке достаточно при нажатой левой кнопке мыши охватить прямоугольником выделения из пунктирных линий требуемый фрагмент и отпустить кнопку мыши. Появится диалоговое окно с размерами выделенного фрагмента (рис. 7.1).

В этом окне можно подправить размеры выделенной части рисунка, например, задать целые значения для отметок координатных осей. Если этого не сделать, у координатных осей появятся довольно длинные и неудобные для применения числа, как это показано на рис. 7.2.

Фактически данная возможность реализует графическую «лупу», позволяющую просматривать небольшие фрагменты графиков. Это полезно для выявления и изучения их особенностей.

Строка **меню окна двухмерной графики**

Работа с кнопками на панели инструментов довольно очевидна. Однако окно двухмерных графиков имеет еще и собственная строка меню, в которой имеются следующие позиции:

File — открытие окна, предварительный просмотр и печать графиков, выход из графического окна (для файлов графиков используется формат **AcroSpin);**

Edit — создание и удаление комментария, стирание последнего графика и копирование графика целиком или выделенной части в буфер обмена;

Set — центрирование, установка масштаба и размера графика;

Plot — инициирование построения графика для выделенного выражения;

Options — установка параметров (см. ниже);

Window — управление окнами;

Help — команды справочной системы Derive.

Команды меню, за исключением команд меню **Options,** в основном аналогичны уже описанным командам меню главного окна. Поэтому ограничимся описанием команд меню **Options:**

Axes — установка параметров координатных осей и надписей у них;

Cross — установка параметров графического курсора (его наличия и цвета);

Grids — установка параметров координатной сетки графика;

Coordinate System — установка системы координат (Rectangular/Polar);

Point — установка параметров точек;

Plot color — установка цвета линий графиков;

Background color — установка цвета фона графиков;

Printing — установка параметров печати;

Follow Mode — установка режима свободного перемещения графического курсора;

Trace Mode — установка режима перемещения курсора по графику;

Autoscal Mode — установка автоперемещения курсора.

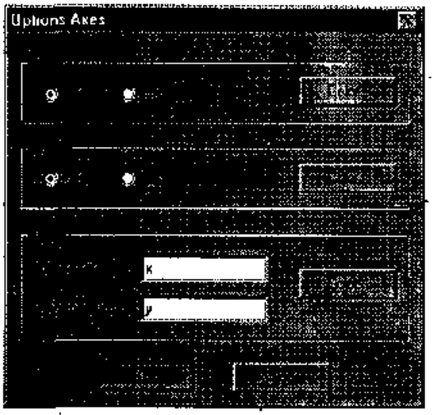


Рис. 7.3. Окно установки параметров координатных осей

Назначение большинства команд очевидно. Некоторые, например команды установки цвета, аналогичны уже описанным. Поэтому остановимся на тех командах, которые достаточно специфичны. После выбора команды **Axes** открывается диалоговое окно, показанное на рис. 7.3.

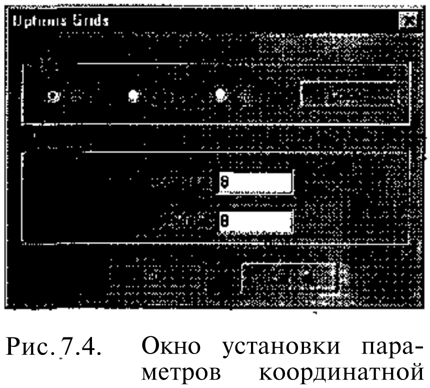


Рис. 7.4. Окно установки параметров координатной сетки

В этом окне в группе переключателей **Lines** можно задать отображение (On) или скрытие (Off) координатных осей, а в группе аналогичных переключателей **Labels** — и меток на них. В группе **Titles** можно задать также надписи у горизонтальной (поле **Horizontal axis)** и вертикальной (поле **Vertical axis)** осей. По умолчанию эти надписи представлены обозначениями осей — буквами x и y. Для всех элементов можно вывести окно установки параметров цвета, используя кнопки **Color.**

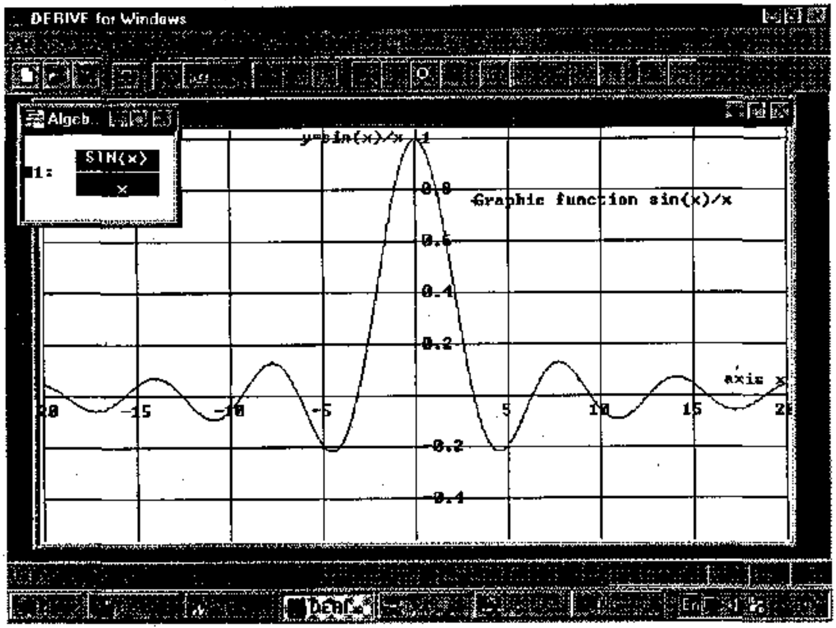


Рис. 7.5. Пример графика

Еще одна команда, которую следует отметить, — Grids, которая выводит окно, позволяющее изменить параметры координатной сетки (рис. 7.4). По умолчанию эта сетка представлена точками — местами пересечения линий сетки.

В этом окне с помощью переключателей группы Display можно задать параметры отображения сетки: в виде точек (Point), линий (Lines) или отсутствие сетки (Off). Кнопка Color позволяет вывести окно установки параметров цвета точек или линий сетки.

На рис. 7.5 показан график функции sin(x)/x при использовании сетки из линий, надписей у координатных осей и комментария Graphic function sin(x)/x.

В целом графики Derive с параметрами, установленными по умолчанию, выглядят вполне привычно. Но, как видно на рис. 7.5, немного поработав над графиками им можно придать более «изысканный» вид.

Графики параметрически заданных функций

Derive позволяет строить графики параметрически заданных функций. Такие функции должны быть попарно записаны в виде вектора, т. е. в виде значений в квадратных скобках, разделенные запятыми. Система Derive автоматически распознает такие функции и строит их график. Перед этим выводится окно установки параметров графиков параметрически заданных функций (рис. 7.6).

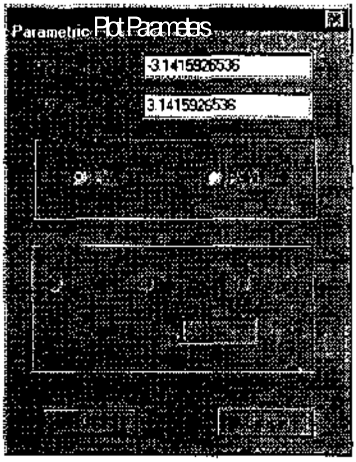


Рис. 7.6. Окно задания параметров параметрически заданных функций

На рис. 7.7 показаны графики двух параметрически заданных функций. Слева выведены окна выражений с заданными в них функциями, а справа — окна с графиками этих функций.

Обратите внимание на график второй функции. Он образован одновременно линией и точками. Параметры отображения графика (см. рис. 7.6) устанавливаются переключателями Line (линия) и Points (точки).

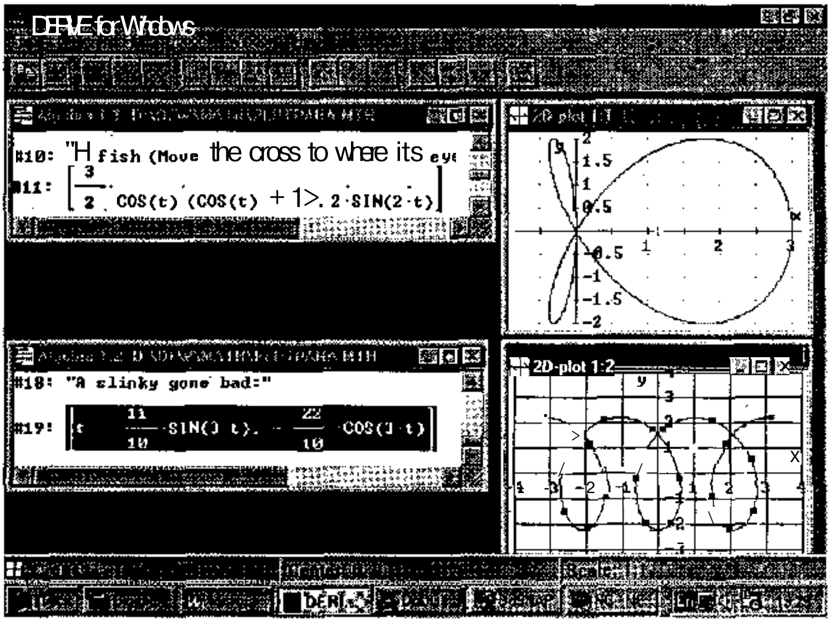


Рис. 7.7. Примеры графиков параметрически заданных функций

Построение графиков функций в полярной системе координат

При построении графика в полярной системе координат надо задать функцию $r(\vartheta)$ и открыть окно двухмерного графика. Кроме того, надо с помощью команды **Coordinate System** установить полярную систему координат (Polar). При инициировании построения графика для выделенного выражения (выборе команды **Plot)** откроется окно **Parametric plot parameter,** в котором можно уточнить параметры построения (если это требуется). Щелчок на кнопке **OK** приводит к построению графика (рис. 7.8).

Назначение кнопок панели инструментов и команд меню в данном случае такое же, как для двухмерных графиков.

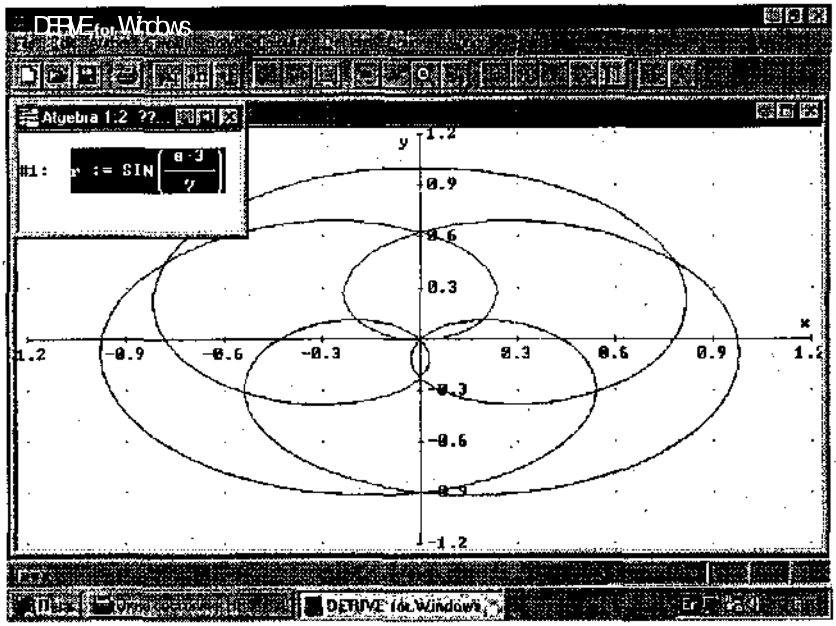


Рис. 7.8. Пример графика в полярной системе координат

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 4.* под Windows введено три режима перемещения графического курсора. Режим **Follow Mode** позволяет просмотреть график функции за пределами видимого окна. В этом режиме, если попытаться вывести курсор за пределы графика, область просмотра (или окно просмотра) скачком смещается в ту часть, куда попал курсор. В режиме **Autoscale Mode** таких скачков нет.

Но особенно интересен режим **Trace Mode,** котором графический курсор превращается из крестика в маленький квадратик и устанавливается на кривую графика, соответствующую выделенному выражению (рис. 7.9). При этом курсор всегда двигается (с помощью мыши или клавиш перемещения курсора) точно по кривой (при нажатой клавише **Ctrl** движение ускоряется). Это повышает возможности анализа особых точек кривых.

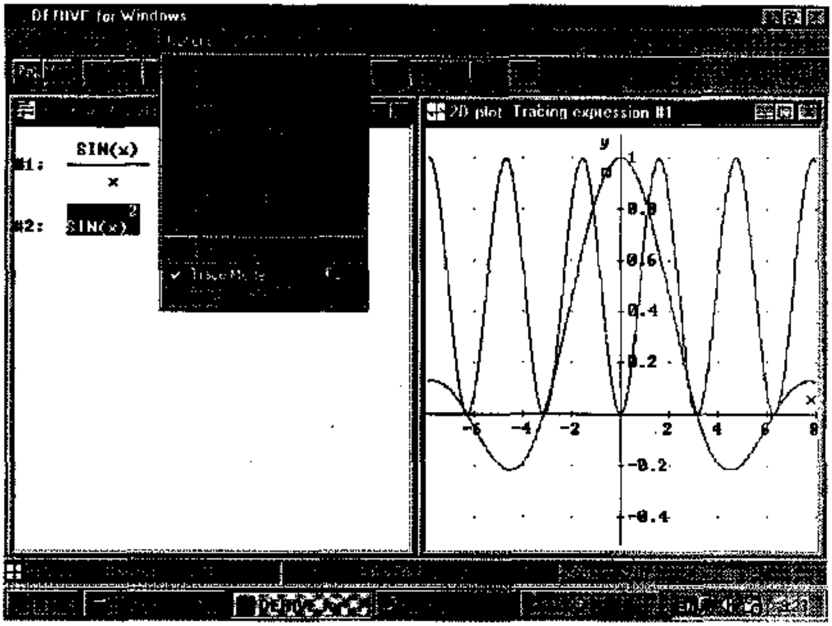


Рис. 7.9. Выбор режима Trace Mode

Графический курсор удобно использовать также для центрирования графика — достаточно поместить его в точку графика, которую желательно сделать центром, и щелкнуть на кнопке центрирования (Center on cross).

Создание нового окна для трехмерной графики

Для построения графиков 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 выбрать команду **New 3D-plot,** которая открывает новое окно для трехмерной графики с собственной строкой меню и панелью инструментов. Эту команду дублирует кнопка **3D-plot window (Ctrl+3)** на панели инструментов главного окна системы.

Чтобы построить график, требуется щелкнуть на аналогичной кнопке на панели инструментов окна графика. Для наглядности можно рядом или поверх него вывести окно с исходным выражением. Осуществив все эти операции, можно получить график, представленный на рис. 7.10.

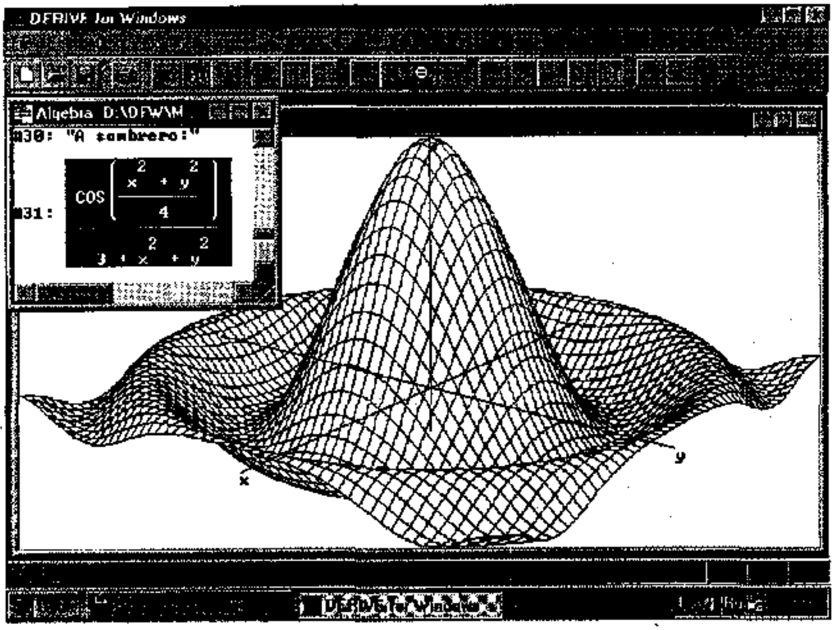


Рис. 7.10. Пример графика поверхности

Нетрудно заметить, что график имеет вид проволочного каркаса, причем линии, которые заслоняет поверхность, скрыты алгоритмом удаления невидимых линий.

Панель инструментов окна трехмерных графиков

С помощью команд панели инструментов или строки меню вид графика можно существенно изменить. Так, на рис. 7.11 показан тот же график, но с другим масштабом и под другим углом зрения. Для изменения угла зрения использована кнопка с изображением глаза.

На рис. 7.11 видны строка меню и панель инструментов окна построения трехмерных графиков. Кнопки панели инструментов перечислены ниже (слева направо).

Копирование и печать графика:

Copy plot window — копирование графика в буфер обмена;

Print — печать графика.

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;

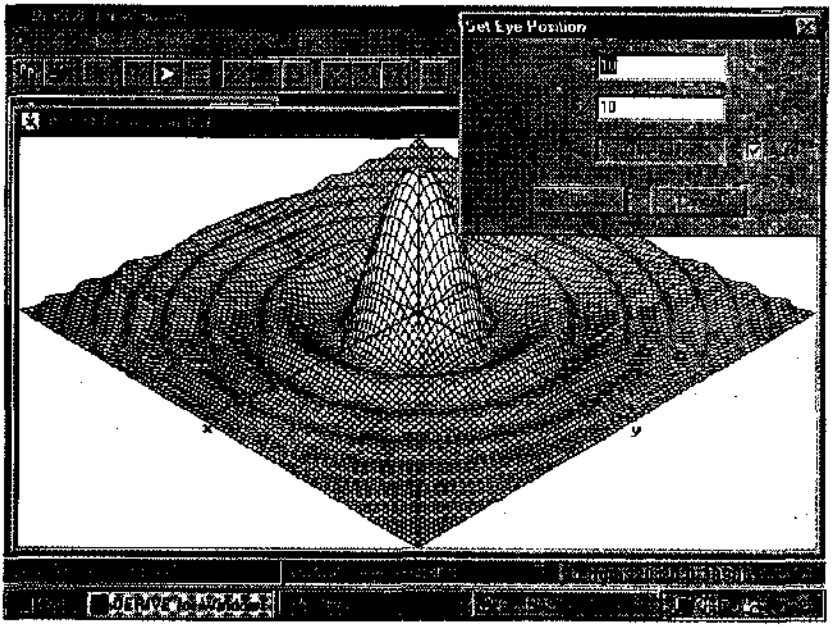


Рис. 7.11. График трехмерной поверхности с измененными масштабом и углом зрения

Zoom vertical out (F8) — расширение области графика по вертикали;

Zoom horizontal out (F6) — 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.

Возвращение в окно выражений **Algebra window (Ctrl+1).**

Лишь команды третьей группы (центрирование и просмотр графика) отличаются от команд окна двухмерной графики. Команда **Set box center** позволяет установить координаты x, y и z центра графика. В режиме **Auto** координата z вычисляется автоматически. Команда **Set viewpoint** выводит окно, показанное в верхнем правом углу рис. 7.11. В нем нужно установить координаты точки, с которой ведется обзор трехмерной поверхности. Для изменения числа линий сетки на трехмерной поверхности служит команда **Set grid panel.**

Строка меню окна трехмерной графики

Команды меню окна трехмерной графики практически аналогичны командам меню окна двухмерных графиков. Поэтому ограничимся описанием только тех команд, которые отличны.

Команды меню **Set:**

Center — установка координат центра;

Eye — установка координат точки обзора;

Grids — установка числа линий сетки по осям x и y;

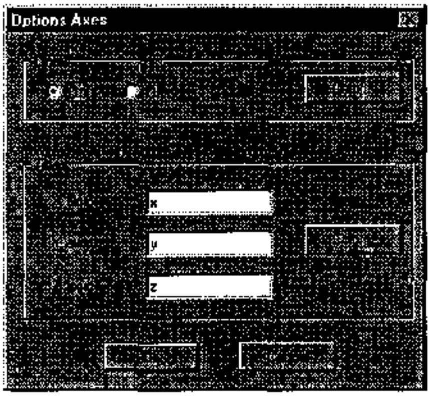


Рис. 7.12, Окно параметров отображения осей Axess трехмерных графиков

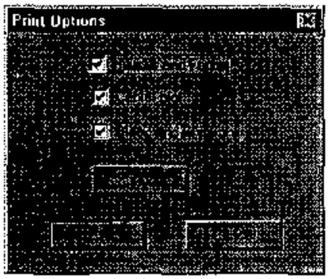


Рис. 7.13. Окно параметров печати

Length — установка интервалов между линиями.

Команды меню **Options:**

Axes — установка параметров координатных осей и надписей и у них;

Plot color — установка цвета линий графиков;

Background color — установка цвета фона графиков;

Printing — установка параметров печати;

Remove Hidden Line — удаление невидимых линий.

Команда **Axess** выводит окно, показанное на рис. 7.12. 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 параметров печати трехмерных графиков (рис. **7.13): Header and Footer** — печать колонтитулов (верхних и нижних), **Center plot** — печать центральной точки графика и **Black and white onle** — печать только в черно-белом цвете. Щелчок на кнопке **Fonts** выводит панель названия доступных шрифтов.

Графика как средство визуализации математических понятий

Графика Derive 4.0 под Windows, несмотря на ее скромные возможности (отсутствие ряда специальных видов графиков, функциональной окраски и т. д.), может служить мощным средством визуализации математических и физических понятий. Рассмотрим примеры.

Понятие о графическом представлении функций весьма важно в курсах математики и физики. На рис. 7.14 показано, как это понятие иллюстрируется визуальными графическими средствами Derive.

Рис. 7.14 дает представление об изменении вида синусоиды при изменении таких ее параметров, как амплитуда, фаза и частота. Особенно ценно, что можно быстро изменить любой параметр и просмотреть, как это скажется на графике функции. На рис. 7.15 наглядно показаны три вида модуляции вы-

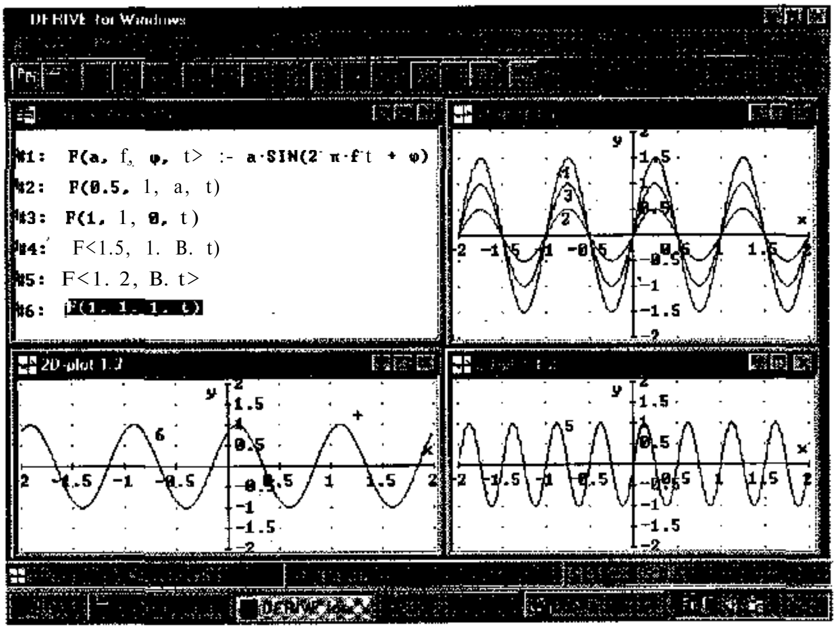


Рис. 7.14. Представление графика синусоиды

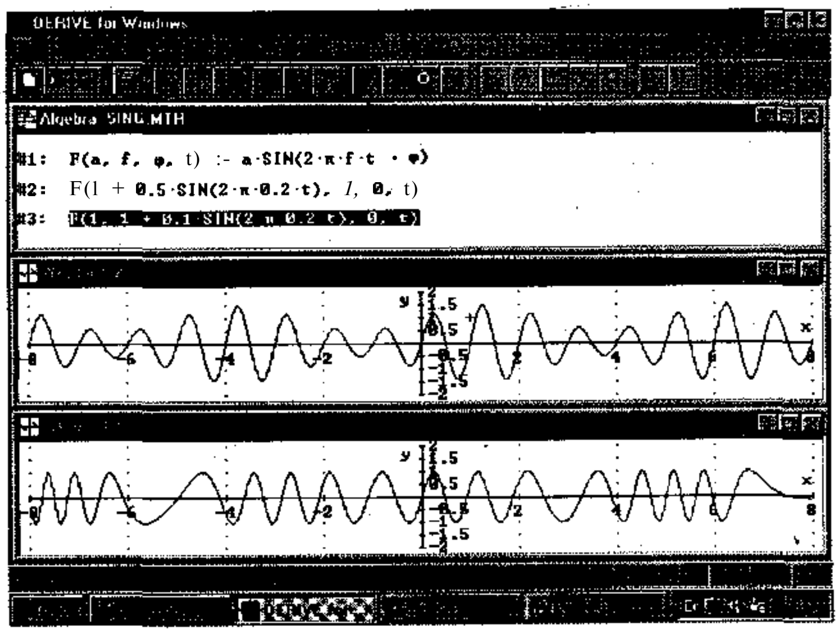


Рис. 7.15. Различные виды модуляции синусоидальных колебаний

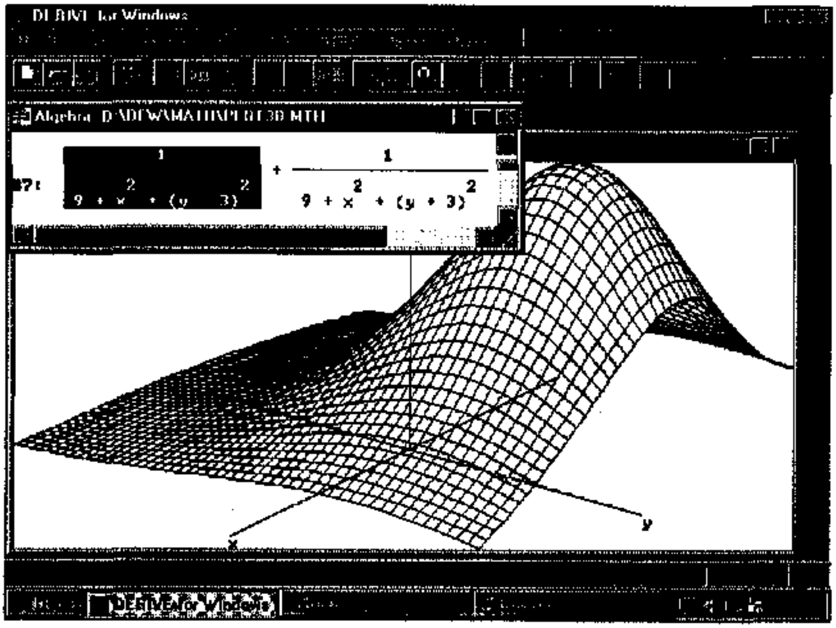


Рис. 7.16. Первая часть трехмерной поверхности

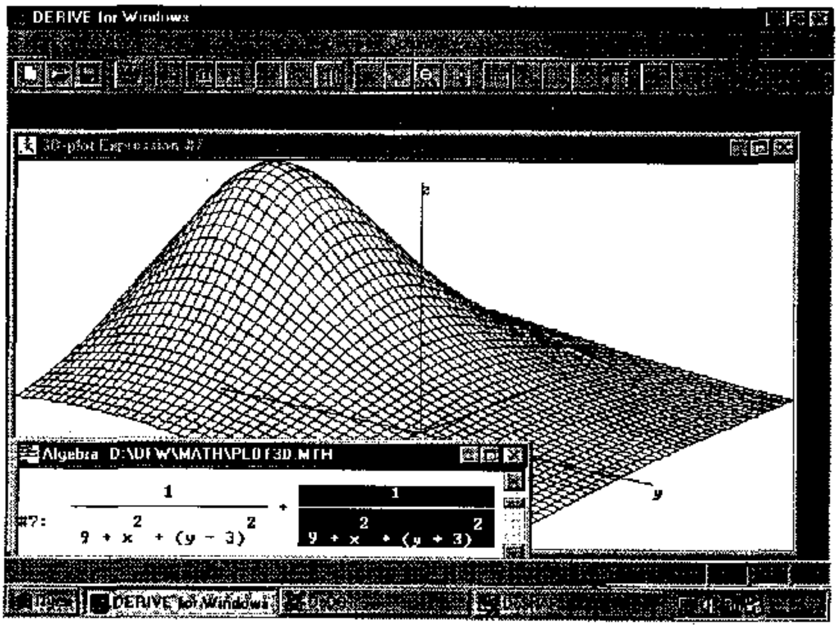


Рис. 7.17. Вторая часть трехмерной поверхности

сокочастотного синусоидального колебания низкочастотным синусоидальным колебанием. Это позволяет понять смысл модуляции и усвоить относящиеся к ней представления.

Большие трудности представляет наглядное представление сложных трехмерных поверхностей, особенно если они описываются составными выражениями. Derive позволяет легко визуализировать такие выражения, строя графики трехмерных поверхностей для их составных частей или для полного выражения. На рис. 7.16 показано некоторое выражение — функция двух переменных, имеющее два члена. Этот рисунок строит поверхность, описанную только одним выделенным членом. Поверхность имеет вид холма, смещенного от центра графика.

Выделив второй член выражения, можно построить соответствующую ему поверхность. Она показана на рис. 7.17 и представляет собой также холм, смещенный относительно холма, показанного на рис. 7.16.

Наконец, на рис. 7.18 показан общий вид трехмерной поверхности, взятой нами для примера. Как видно, эта поверхность имеет два холма, плавно переходящие друг в друга. Языком математики можно сказать, что она является линейной комбинацией двух поверхностей, каждая их которых описывается своим выражением.

Можно привести массу и других примеров наглядной визуализации математических понятий. Прекрасный пример этого — разложение функций в ряд Тейлора (рис. 6.35). Множество примеров визуализации аналитических вычислений было описано выше.

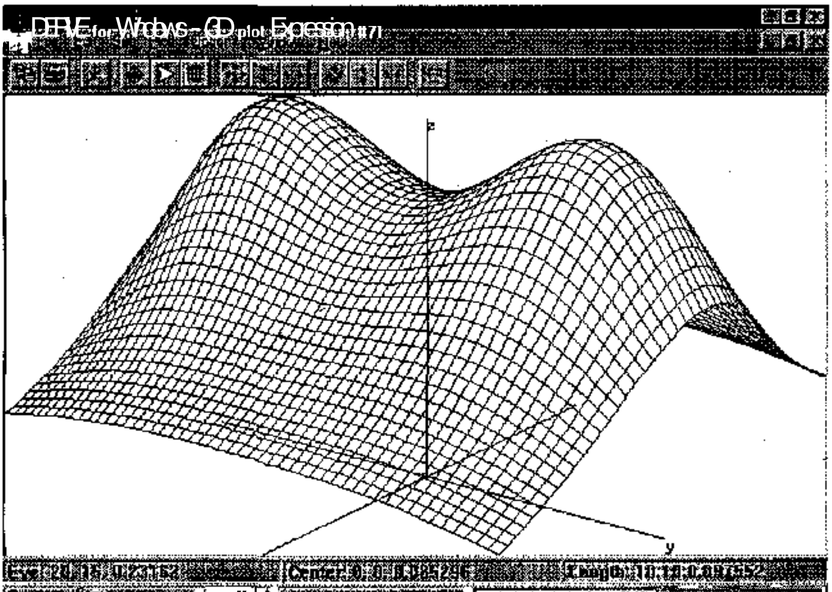


Рис. 7.18.Общий вид трехмерной поверхности типа «два **холма»**

Одновременное построение ряда графиков в одном окне

Derive 4 под Windows позволяет строить несколько графиков функций в одном графическом окне путем последовательного построения раз за разом графиков, выделяемых по одному из выражений (см. пример на рис. 6.35).

Однако в ряде случаев желательно одновременное построение сразу ряда графиков функций в одном окне при задании их списком. Derive 4 под Windows обеспечивает такую возможность для двухмерных графиков. Для этого достаточно набрать нужные выражения как функции одной переменной в квадратных скобках, выделить полученный список и использовать обычные операции построения 3D-графиков. Рис. 7.19 поясняет такое построение.

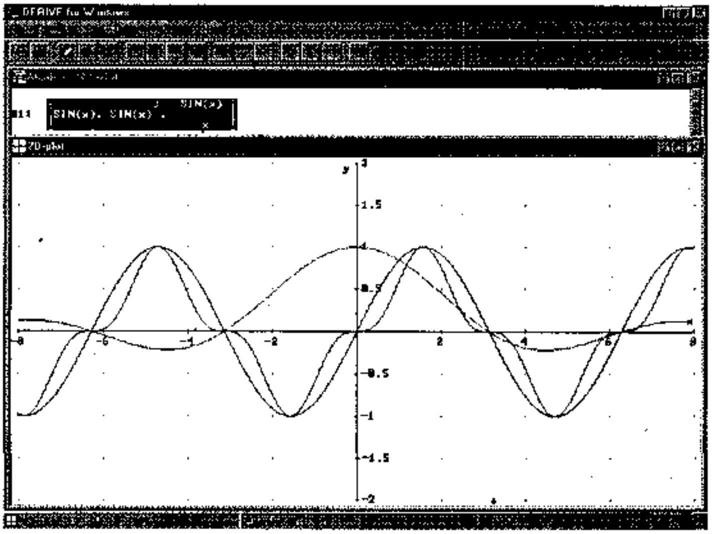


Рис. 7.19. Построение графиков ряда выражений, представленных списком

Derive автоматически устанавливает разные цвета для разных кривых. Форматирование графика такого вида производится стандартными средствами форматирования двухмерных графиков, описанными выше.

Справочная **система**

Отличительной чертой Derive под Windows стала достаточно мощная справочная система, удовлетворяющая всем стандартам таких систем для Windows-приложений. Для работы со справочной системой Derive служит меню **Help** со следующими командами:

Contents — вызов справки по контексту;

Index — вызов индексного каталога;

Using Help — вызов справки о самой справочной системе;

About Derive — вывод информационного окна **с** краткими сведениями о системе.

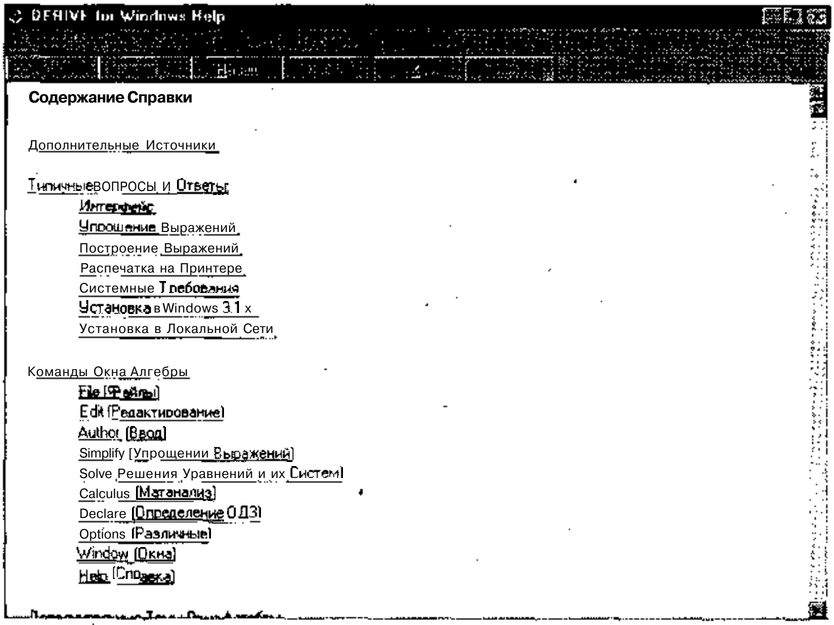


Рис. 7.20. Окно справки по контексту системы Derive 4.* под Windows

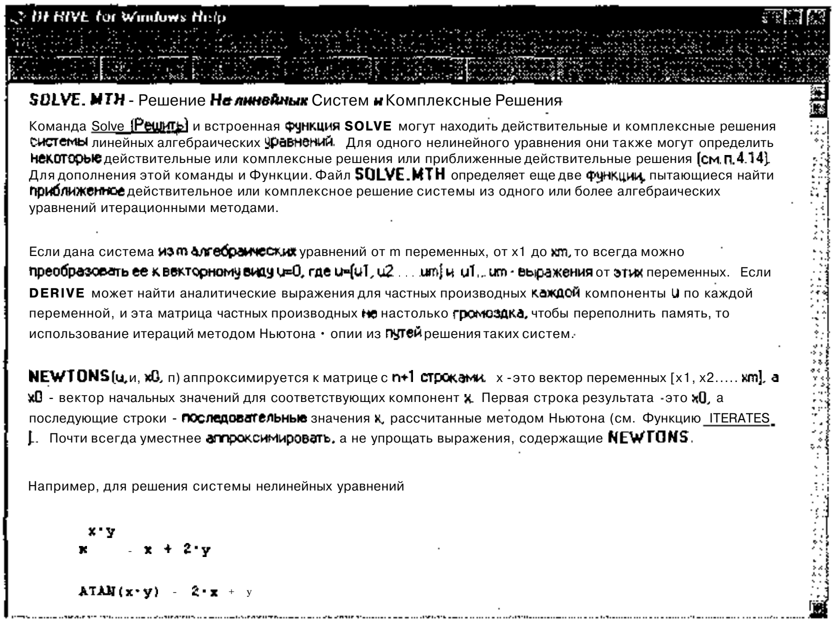


Рис. 7.21. Пример предоставления справки по пакету расширения

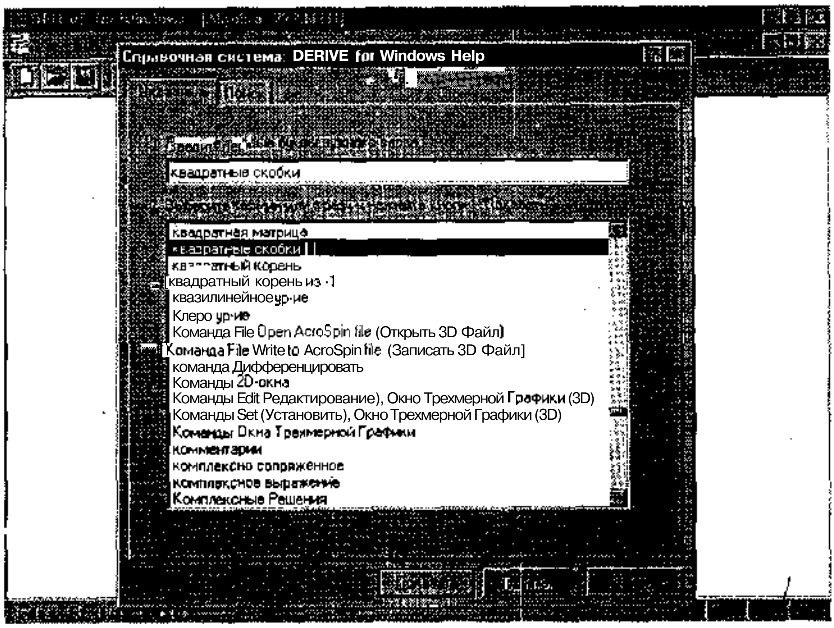


Рис. 7.22. Окно индексного каталога с открытой вкладкой Указатель

Поскольку большинство наших читателей испытывают затруднение с применением англоязычной справки, ниже дается краткое описание справочной системы русифицированной версии Derive 4.* под Windows. Основное окно справочной системы показано на рис. 7.20. Оно появляется при исполнении команды **Help Contenst.**

Для вывода справок используется стандартное окно справок операционной системы Windows. Если на ПК установлена русифицированная операционная система, то надписи на элементах этого окна будут на русском языке.

В отличие от Derive под MS-DOS, справки в версиях Derive под Windows даны достаточно подробные с примерами применения операторов и функций. В качестве иллюстрации к этому на рис. 7. 21 представлен образец вывода справки по пакету расширения SOLVE.MTH, предоставляющему дополнительные функции для решения уравнений и систем уравнений.

Уже из начала этой справки видно, что дается не только краткое определение функций данного пакета расширения, но и примеры их применений. Это значительно облегчает знакомство с возможностями системы Derive под Windows. Впрочем, надо отметить, что, в отличие от справок современных, более мощных систем компьютерной математики (например, Mathcad или Mathematica) примеры в справочной системе Derive под Windows не являются «живыми», т. е. их нельзя модифицировать прямо в справке и тут же исполнить с новыми данными.

Команда **Help Index** выводит окно индексного каталога, показанное на рис. 7.22. В этом окне имеется две вкладки — **Указатель** для поиска темы справки по заданному ключевому слову или ключевой фразы в окне 1

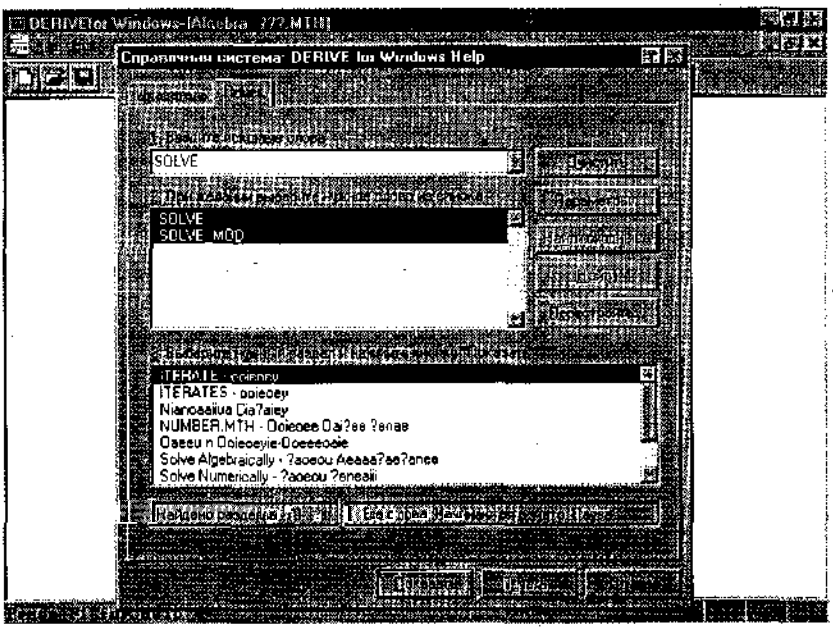


Рис. 7.23. Окно индексного каталога с открытой вкладкой Поиск

(см. пример на тему «квадратные скобки») и **Поиск** для поиска всех тем, в которых упоминается ключевая фраза или слово.

В процессе ввода ключевой фразы или слова в поле раздела 1 в окне 2 выделяется название соответствующего раздела справки. Для вывода страницы справки с этим разделом достаточно активизировать мышью кнопку **Показать окна индексного указателя.** Кнопка **Печать** позволяет распечатать справку, а кнопка **Отмена** отказаться от работы со справкой.

Вид окна индексного каталога с открытой вкладкой **Поиск** представлен на рис. 7.23. В этом случае могут быть найдены все разделы справочной системы, в которых упоминается ключевое слово или фраза (см. пример на поиск по ключевому слову SOLVE).

К сожалению, даже в Derive 4 под Windows есть проблемы с русификацией справочной базы данных, связанные с применением некоторых наборов шрифтов, которые есть не на всех компьютерах. Это хорошо видно из надписей в окне 3 вкладки Поиск, где часть русскоязычных надписей явно не читается. Естественно, что в оригинальной англоязычной версии Derive таких проблем нет.

Команда **Help — Using Help** выводит основное окно справки уже по самой справочной системе (рис. 7.24).

Знакомство с применением справочной системы полезно разве что пользователям, впервые осваивающим не только систему Derive, но и свой недавно приобретенный компьютер. Большинство пользователей ПК прекрасно знают, как пользоваться типовой справочной системой приложений под Win-

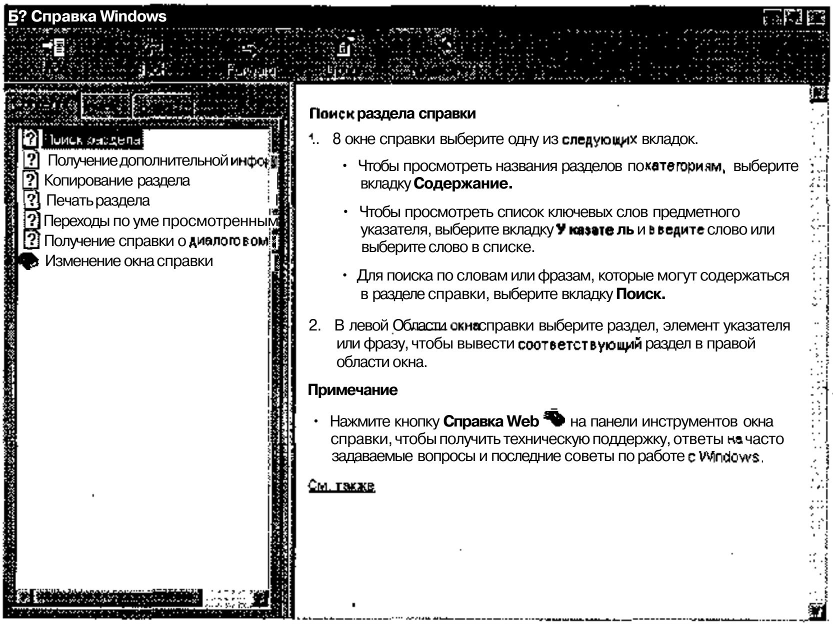


Рис. 7.24. Окно справки по справочной системе

dows. Поэтому воздержимся от более подробного описания справочной системы Derive под Windows.

Последняя команда **About DERIVE...** в меню **Help** выводит информацию о системе Derive и ее разработчике в виде красочной заставки, которая подобна приведенной на рис. 6.1.

Чему мы научились:

- Устанавливать параметры системы.
- Работать с окнами.
- Использовать графические средства 2D- и 3D-графики.
- Пользоваться справочной системой.

Урок 8. Derive 5 под Windows

- Установка и запуск Derive 5 под Windows
- Новые возможности Derive 5
- Расширенные возможности редактирования
- Операции вставки
- Установка опций Derive 5
- Вкладки окон открытых документов
- Новые возможности графики Derive 5
- Управление средствами графики
- Новые возможности в вычислениях

Установка и запуск Derive 5 под Windows

Derive 5 под Windows — новейшая и последняя на момент подготовки данной книги версия системы Derive. В ней предпринята успешная попытка преодоления главных недостатков всех предшествующих версий этой популярной системы — ущербности интерфейса пользователя и ограниченных возможностей в создании документов класса ноутбуков и графики.

Derive 5 под Windows поставляется на CD-ROM и инсталлируется с него. Инсталляция не имеет каких-либо особенностей по сравнению с инсталля-

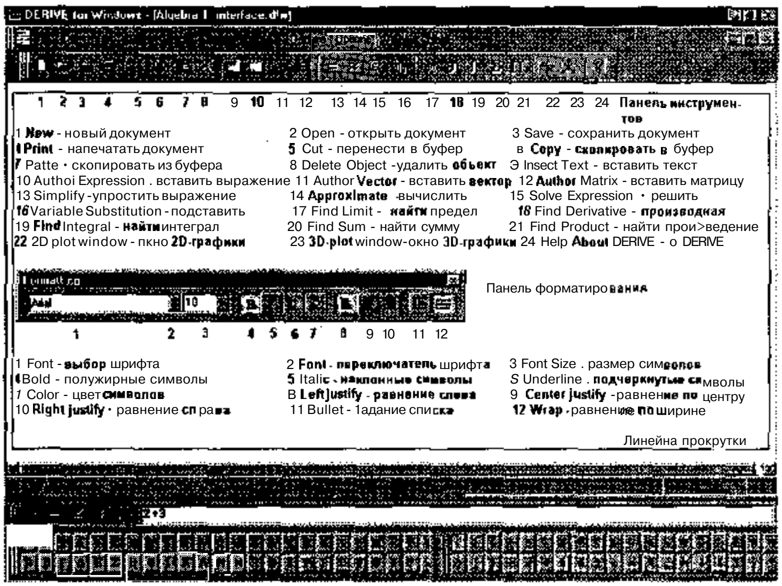


Рис. 8.1. Окно Derive под Windows с элементами интерфейса и указанием их назначения цией большинства приложений под Windows. Поэтому на ней мы останавливаться не будем. При возникновении затруднений с инсталляцией, как обычно, надо ознакомиться с файлом readme.

Для нормальной работы системы требуется ПК с объемом оперативной памяти, достаточным для установки операционной системы Windows 95/98/NT. Б отличие от предшествующих версий, Derive for Windows 5 работает только на 32-разрядных ПК, работа на 16-разрядных ПК не поддерживается. Для установки системы в минимальном варианте нужно около 3 Мбайт памяти на жестком диске, так что Derive 5 под Windows по-прежнему является довольно маленькой программой.

При запуске Derive 5 (как обычно, из меню программ или активизацией ярлыка системы) появляется окно системы, которое в полностью развернутом виде представлено на рис. 8.1.

В ряде позиций меню **(File, Author, Simplify, Solve, Calculus** и **Declare)** вы не заметите ничего нового — их команды и опции те же, что были описаны в уроке 6, посвященном описанию предшествующей версии Derive 4.* под Windows. Поэтому мы не будем описывать эти команды и отсылаем читателя к указанному материалу. Однако в других позициях меню можно найти множество новых команд, открывающих новые возможности системы Derive 5 под Windows (или просто Derive 5). Ниже мы их рассмотрим вначале кратко, а затем детально.

Новые возможности Derive 5

Краткий обзор новых возможностей

Из рис. 8.1 видно, что вид интерфейса системы существенно изменился, причем явно в лучшую сторону. Теперь система имеет стандартные титульную строку, меню, панель инструментов и панель форматирования, обычно располагаемые сверху. При этом все эти элементы (кроме титульной строки) могут перемещаться мышью -в любое место экрана и занимать положение, наиболее удобное для пользователя (см., например, панель форматирования на рис. 8.1, помещенную в центр окна).

Внизу окна расположена, как обычно, строка состояния (статуса) системы. А вот и новинка — по умолчанию под строкой статуса выводится строка ввода выражений **Author** и панель ввода специальных символов и основных операторов. Это существенно облегчает ввод выражений и уменьшает необходимость в использовании панели инструментов и меню системы.

Рис. 8.2 показывает решение кубического уравнения в среде Derive 5 под Windows. На этот раз панели ввода спецсимволов и операторов удалены, а строка ввода размещена под панелью инструментов. Если отвлечься от этих деталей улучшенного интерфейса, то станет ясно, что работа с Derive 5 под Windows ничем не отличается от описанной выше работы с Derive 4.* под Windows. Вид окна документов при обычных вычислениях одинаков у обоих систем.

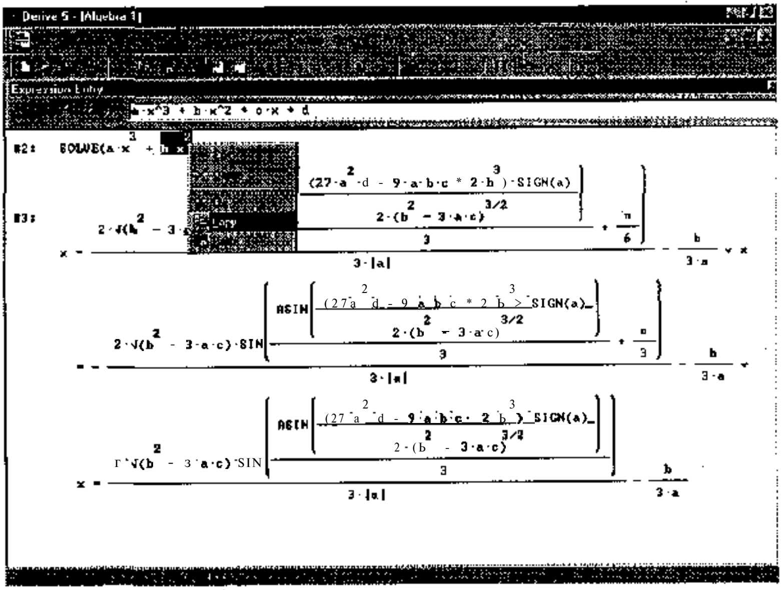


Рис. 8.2. Пример решения кубического уравнения в символьном виде

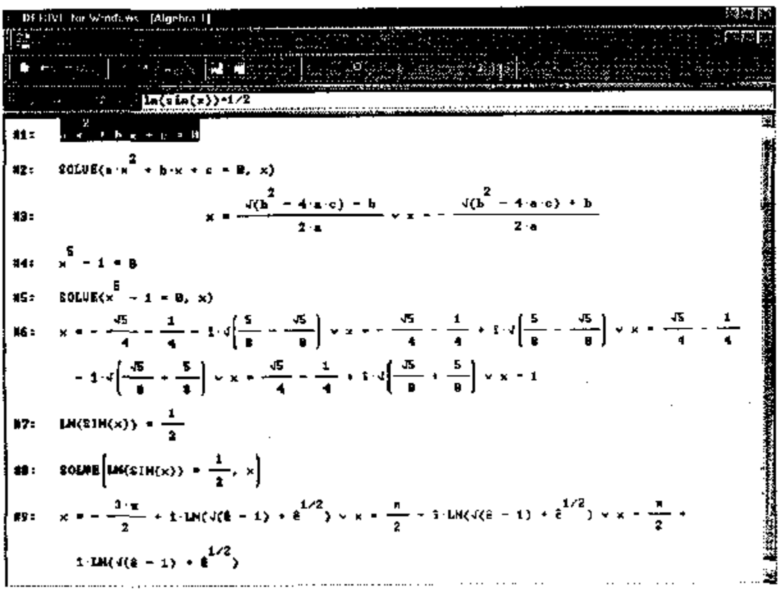


Рис. 8.3 Примеры типовых вычислений в среде Derive 5 под Windows

К новым возможностям Derive 5 относится возможность выделения частей выражений указанием на них курсора мыши и двойным щелчком левой клавиши, а также полноценные функции правой клавиши мыши. Теперь нажатие правой клавиши ведет к появлению контекстно зависимого

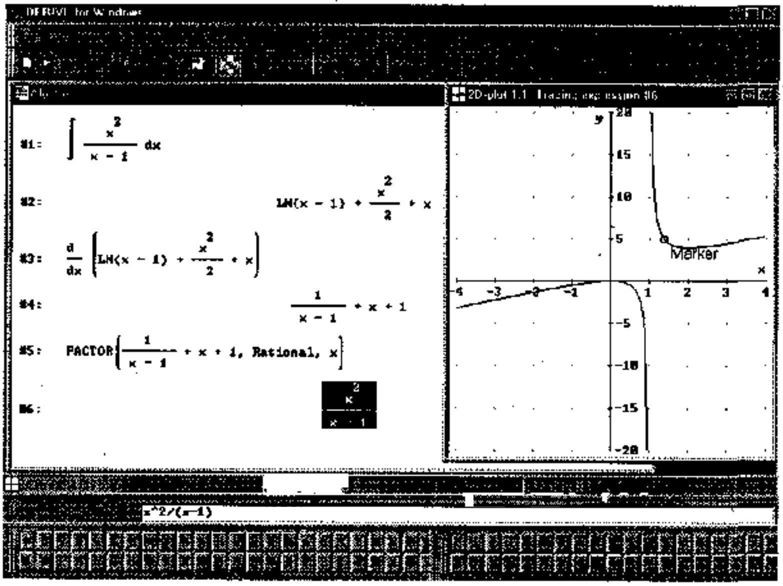


Рис. 8.4. Примеры вычислений и построения графика выражения

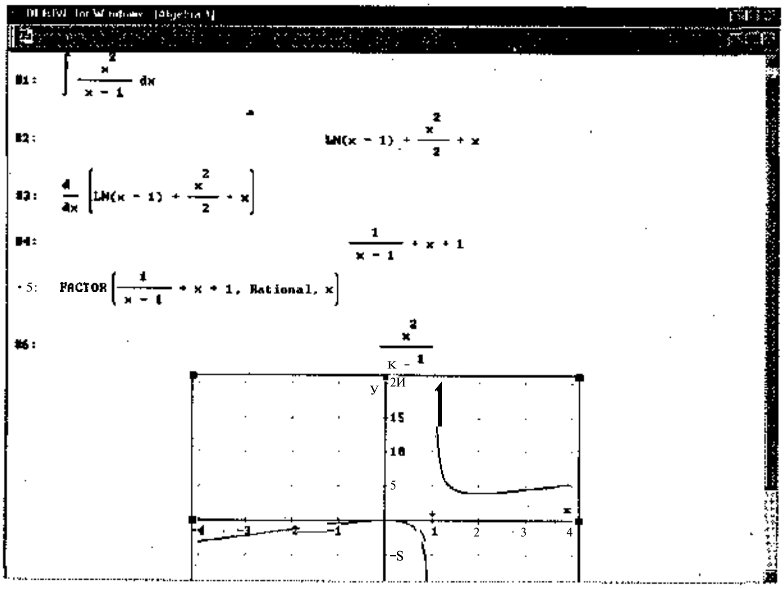


Рис. 8.5. Пример построения графика в поле документа

меню (оно видно на рис. 8.2, как и выделение части выражения), в котором присутствуют те команды, которые допустимы для выделенного объекта. Строку ввода можно постоянно держать в окне документа, что упрощает и ускоряет ввод выражений. Доступны все типовые операции с буфером обмена (в том числе и через контекстно зависимое меню правой клавиши мыши).

Рис. 8.3, на котором показаны типовые примеры вычислений в среде Derive 5 под Windows, показывает, что и в русифицированной операционной системе Derive 5 не искажает символы мнимой единицы, квадратного корня и основания натурального логарифма.

Вполне корректно Derive выводит и знаки интегралов и оператора дифференцирования (рис. 8.4). Как и предшествующая версия, Derive 5 позволяет строить графики выделенных выражений, выводя их в специальное окно графики. Обратите внимание на то, что если окно графики активно, то панель инструментов окна документов заменяется панелью инструментов графического окна. Графическое окно может мышью плавно меняться в размерах и перемещаться.

А вот рис. 8.5 демонстрирует новую возможность Derive 5 под Windows — построение графиков не в отдельных окнах, а прямо в поле документа. Это, несомненно, первый крупный шаг в создании документов в стиле ноутбуков (блокноты). Напоминаем, что документы в таком стиле предполагают вывод в документе одновременно математических выражений, графиков и текстовых надписей без ограничения размеров и с разным стилем. Для реализации этой возможности нужно пользоваться командой **Insert 2D-plot Object** в случае двухмерного графика и **Insert 3D-plot Object** в случае трехмерной графики.

Важно отметить, что документы в стиле ноутбуков сохраняются при использовании команд **Save** и **Save As...** в позиции **File** меню. Таким образом, Derive 5 избавился от такого порока прежних систем, как невозможность сохранения одновременно математических выражений с текстовыми комментариями и графиков.

Обратите внимание, что если график выделен мышью, то он обрамляется характерной рамкой из темных линий с местами установки курсора мыши. Установив в эти места курсор мыши, можно растягивать график в разных направлениях. К сожалению, качество представления графиков после такой операции заметно ухудшается, особенно при увеличении размеров графиков.

Следующий шаг в превращении документов в ноутбуки заключается в возможности ввода текстовых комментариев различного стиля. Derive 5 предусматривает возможность создания текстовых областей без номеров, характерных для областей ввода и вывода. Это и придает документам вид вполне полноценных ноутбуков (см. рис. 8.6 с такими текстовыми областями).

Рис. 8.6 заодно демонстрирует и еще одну новинку в новой версии Derive — возможность построения трехмерных графиков с цветной функциональной окраской поверхностей или фигур. При этом такое построение обеспечивается по умолчанию и существенно повышает наглядность трехмерных графиков.

Говоря о графиках, важно отметить, что Derive 5 позволяет записывать графики в файлы различного формата: DIB, JPEG, PCX, TARGA и TIF. Это делается с помощью команды **File Export** меню графики (кстати говоря, меню модифицируется в зависимости от того, какое окно — выражений или графики — активно). Весьма полезной является и опция **Simplify Before Ploting,** которая введена в подменю опций **Options** меню графического окна.

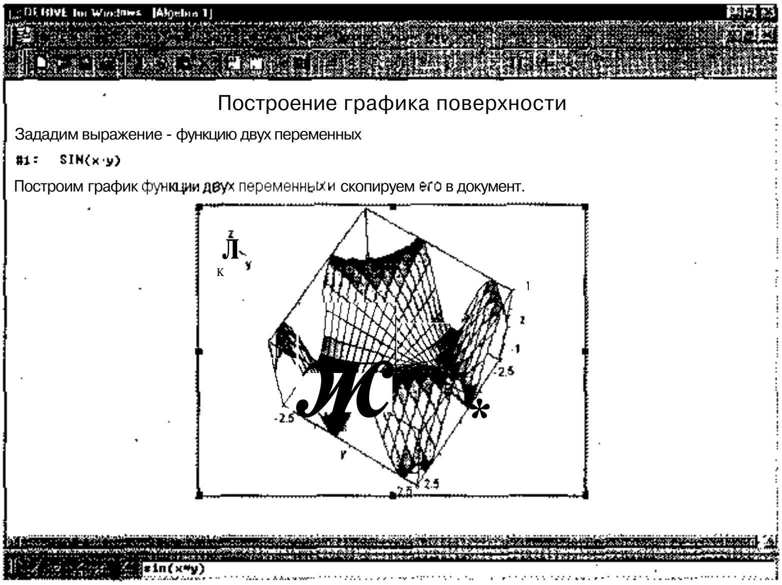


Рис. 8.6. Пример построения документа в стиле ноутбука

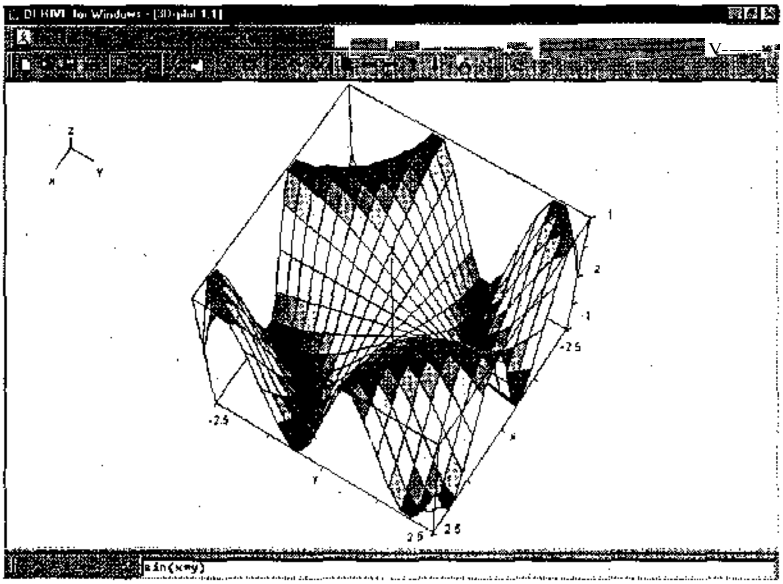


Рис. 8.7. Пример модификации графического трехмерного объекта

Эта опция при ее введении обеспечивает вычисление выражения перед построением графика.

Возможность модификации графиков, имеющаяся в предшествующей версии, сохранена и существенно дополнена. Как видно из рис. 8.7, в панели инструментов графики появились новые кнопки, позволяющие плавно вращать трехмерный объект графики вокруг горизонтальной или вертикальной осей, а также приближать или удалять объект.

В дальнейшем мы отметим и другие нововведения в систему Derive 5 под Windows. Но даже из описанного ясно, что сделан крупный шаг в превращении системы во вполне полноценную систему компьютерной математики если и не высшего, то среднего класса. А если учесть, что при этом Derive осталась системой с весьма скромными аппаратными запросами, то становится ясно, что позиции этой системы (особенно в сфере образования) существенно укрепились.

Расширенные возможности редактирования

Позиция Edit меню

Как видно из рис. 8.8, позиция Edit меню в Derive 5 подверглась существенной модификации.

Она стала похожа по набору команд на типичную позицию Edit большинства приложений под Windows.

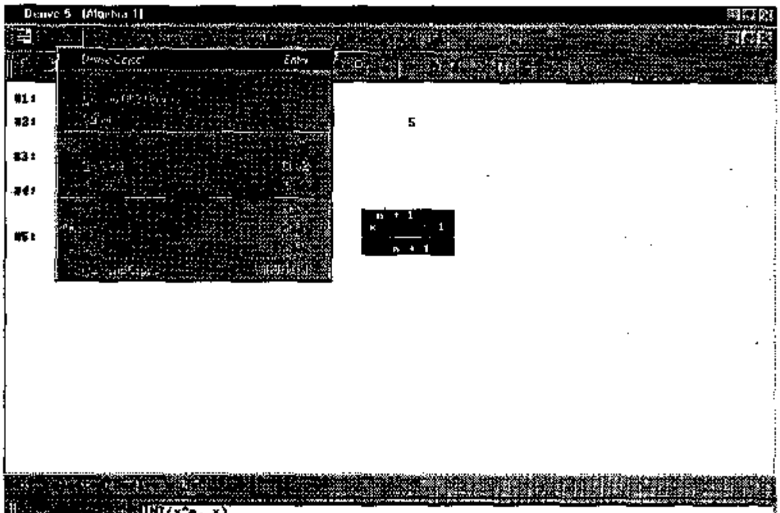


Рис. 8.8. Окно Derive 5 с подменю позиции Edit меню

Ввод выделенных выражений в поле редактирования

Рассмотрим последовательно возможности редактирования в системе Derive 5. Команда Derive **Object...Enter** дает простую возможность копирования выделенного выражения в поле редактирования. Теперь, выделив объект

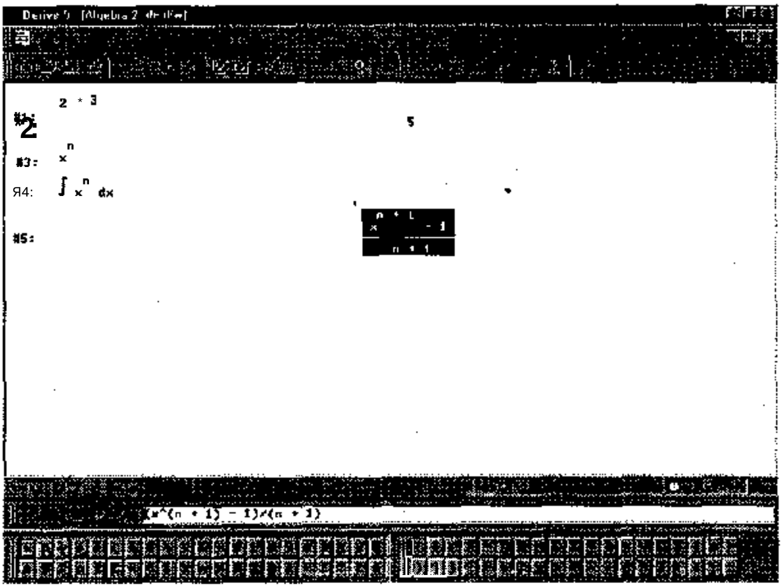


Рис. 8.9. Пример копирования выделенного объекта в поле редактирования

Derive (такой объект виден на рис. 8.8) для его ввода в поле редактирования, достаточно нажать клавишу **Enter** или исполнить данную команду. Рис. 8.9 показывает результат этого.

Внимание

Обратите внимание, что в общем случае стиль выражения при таком копировании меняется. Выражение в документе имеет близкую к математической форму, тогда как в поле редактирования ***оно*** *превращается в однострочное выражение, имеющее форму, допустимую для строчного редактора.*

Задание и просмотр аннотаций к объектам

Следующая новая позиция подменю **Edit — Annotation** позволяет задать аннотацию (текстовое пояснение) для выделенного объекта. Создание аннотации поясняет рис. 8.10. На нем показано окно ввода аннотации, которое появляется при исполнении этой команды.

По умолчанию в поле ввода аннотации обычно имеется текстовая надпись, но пользователь может заменить ее своей. К сожалению, без перестройки наборов используемых шрифтов аннотацию нельзя задавать символами кириллицы. После ввода аннотации она просматривается в строке состояния системы, если выделен объект, к которому относится аннотация (рис. 8.11).

Как уже отмечалось, ввод аннотаций позволяет идентифицировать объекты, например, отметить, является ли объект исходным выражением или результатом тех или иных его преобразований.

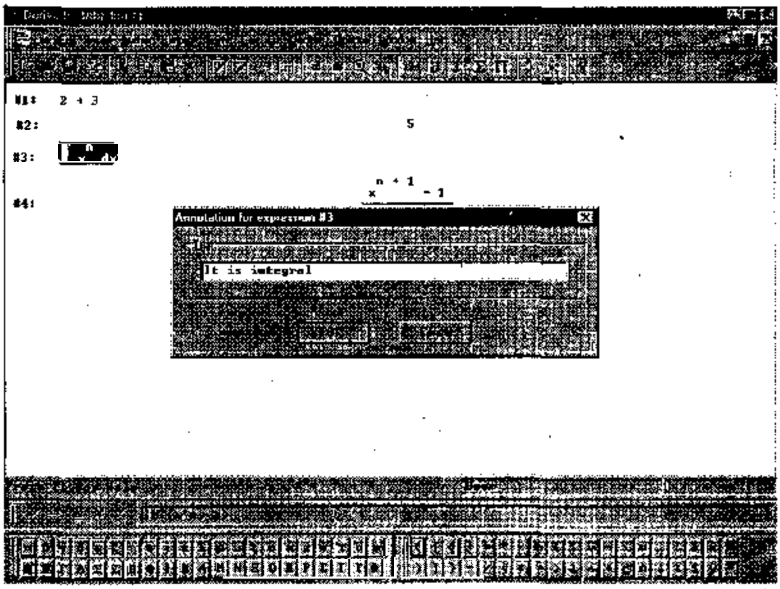


Рис. 8.10. Задание аннотации к выделенному выражению

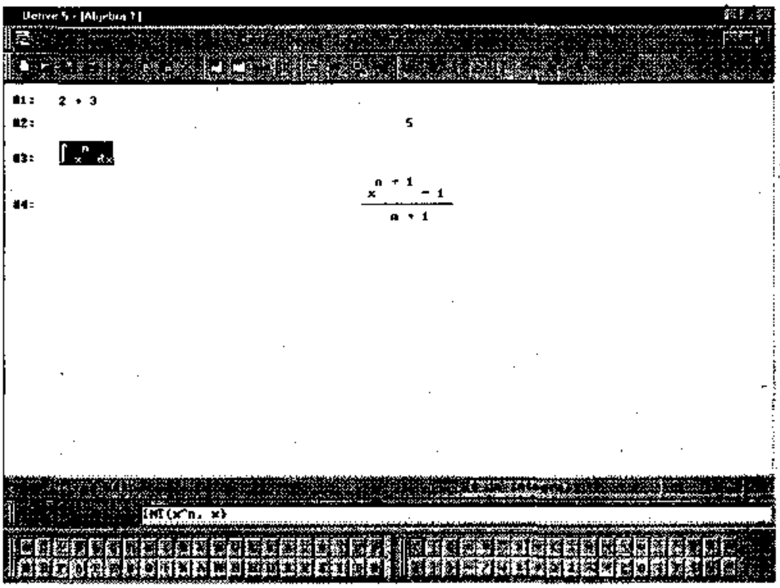


Рис. 8.11. Пример просмотра аннотации к выделенному объекту

Две следующие новые позиции подменю **Edit** служат для установки связи с объектом **(Links OLE Objects...)** и вызова объекта на редактирование **(Объект).** Они будут обсуждены позже, после описания вставки объектов. Приятно отметить, что в этом подменю появились традиционные (но ранее в Derive 4 под Windows отсутствующие) команды — **Cut Ctrl+C** (вырезка выделенного объекта с его удалением из документа) и **Paste Ctrl+V** — вставка объекта из буфера в документ (Paste). Команда **Copy Expression** переименована в **Copy**, а команда **Mark and Copy** осталась без изменения. Таким образом, новая версия Derive приобрела возможности стандартных операций с буфером промежуточного хранения.

Операции вставки

Вставка текстовой области

Как отмечалось выше, Derive приобрел новую возможность — вставки в документ полноценных по стилю и оформлению текстовых блоков. Для этого в подменю вставки **Insert** появилась новая позиция **Text Object... F5. В** образовавшийся прямоугольник можно вводить текстовые надписи с различным стилем и размером символов и разным их цветом (рис. **8.12).**

Форматирование надписей осуществляется стандартной панелью форматирования. Кроме указанных возможностей, возможно выравнивание надписей в текстовом блоке по правой или левой стороне, а также их центровка.

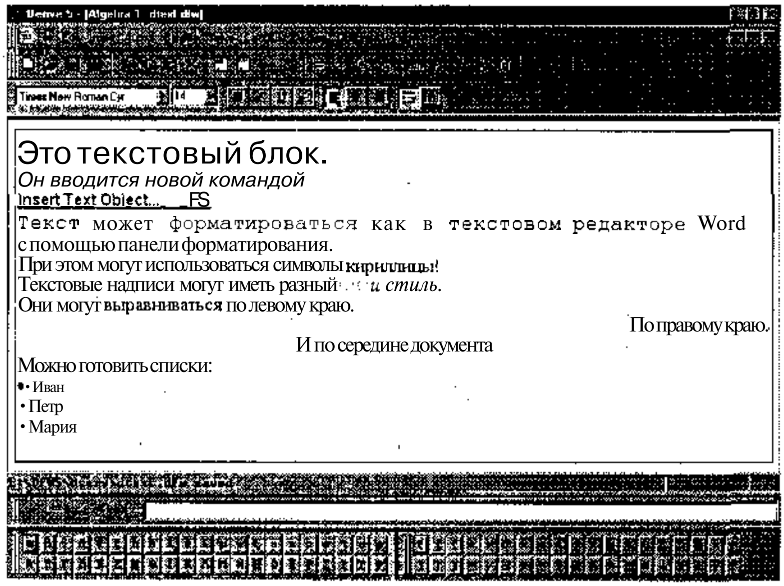


Рис. 8.12. Создание текстовой области в документе

Вставка и редактирование объектов

Вставка объектов — операция, хорошо известная пользователям всех серьезных приложений под Windows. Она впервые появилась в Derive 5 под Windows и реализована командой **OLE Object...** в позиции **Insert** меню. При

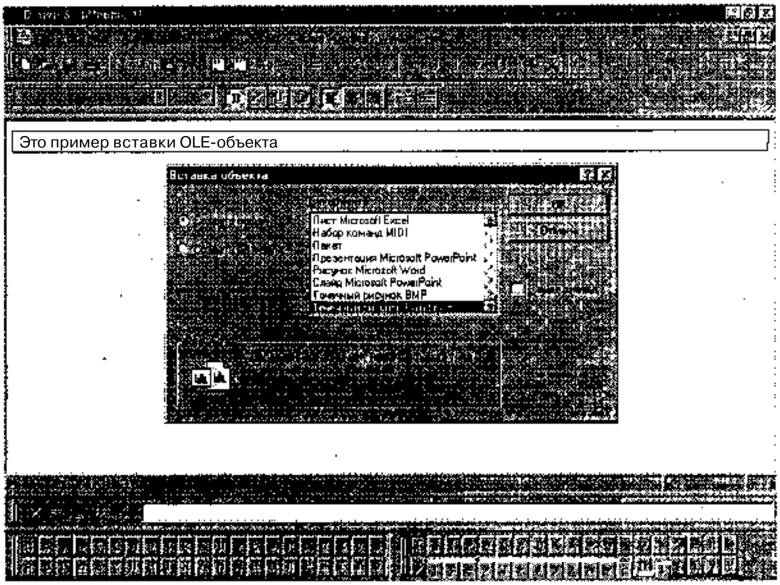


Рис. 8.13. Подготовка к вставке OLE-объекта

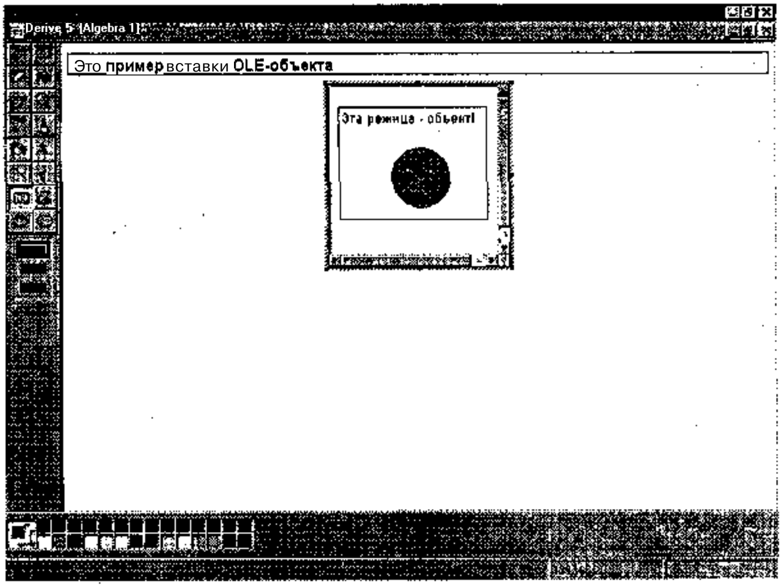


Рис. 8.14. Подготовка объекта в редакторе Paint, вызванном из Derive

исполнении этой команды появляется окно вставки объектов, представленное на рис. 8.13.

В этом окне имеется обширный список приложений, которые могут создавать объекты. Выберем одно из предложений — рисунки, создаваемые графическим редактором Paint, входящим в состав операционной системы Win-

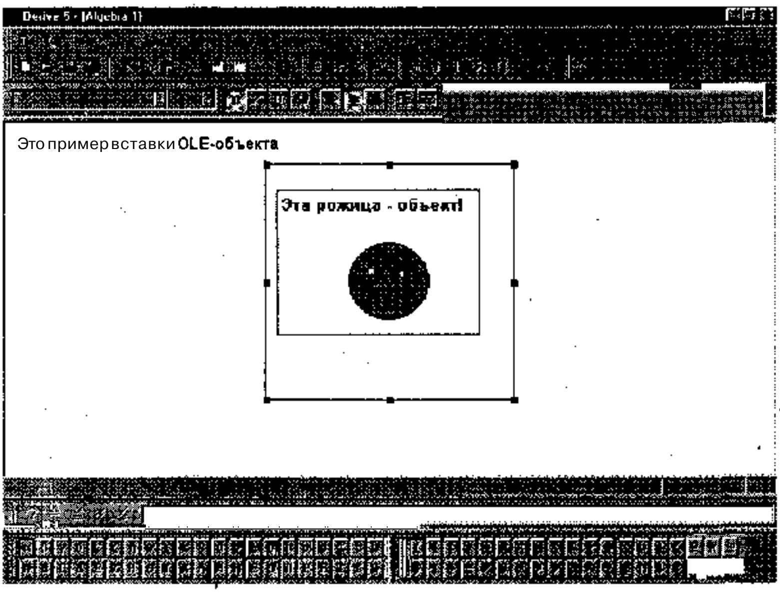


Рис. **8.15.** Результат вставки объекта — рисунка

dows. Нажатие кнопки OK окна приведет к загрузке этого редактора, что будет хорошо видно по изменению вида интерфейса Derive (рис. 8.14). В окне Derive появятся характерные детали интерфейса графического редактора Paint, в частности, его панель инструментов и панель с палитрой цветов.

Теперь можно приступить к подготовке встраиваемого объекта, в частности рисунка. Выйдя из графического редактора, например, щелчком левой клавиши мыши за пределами окна с редактируемым объектом, можно наблюдать появление объекта — рисунка в документе Derive (рис. 8.15). Этот объект имеет тот же вид, что в конце редактирования, но его (пока он выделен рамкой) можно растягивать мышью в разные стороны, цепляясь ее курсором за черные квадратики на сторонах и в углах выделяющей рамки.

Как известно, главная особенность механизма объектной вставки OLE заключается в установлении объектной связи между объектом и создавшим его приложением — в нашем случае графическим редактором Paint. При этом для изменения вида объекта достаточно навести на него курсор мыши и дважды быстро щелкнуть левой ее клавишей. Тут же автоматически запустится создавшее объект приложения, и мы получим вид окна Derive, уже представленный рис. 8.14. После этого можно редактировать объект, добавляя в него новые детали или модернизируя уже имеющиеся.

Преобразование **представления объектов**

Помимо вставки объекта, механизм OLE связи допускает представление объекта еще и в виде значка (пиктограммы) некоторого обобщенного вида. Ниже мы рассмотрим такую вставку. А пока отметим, что Derive 5 позволяет

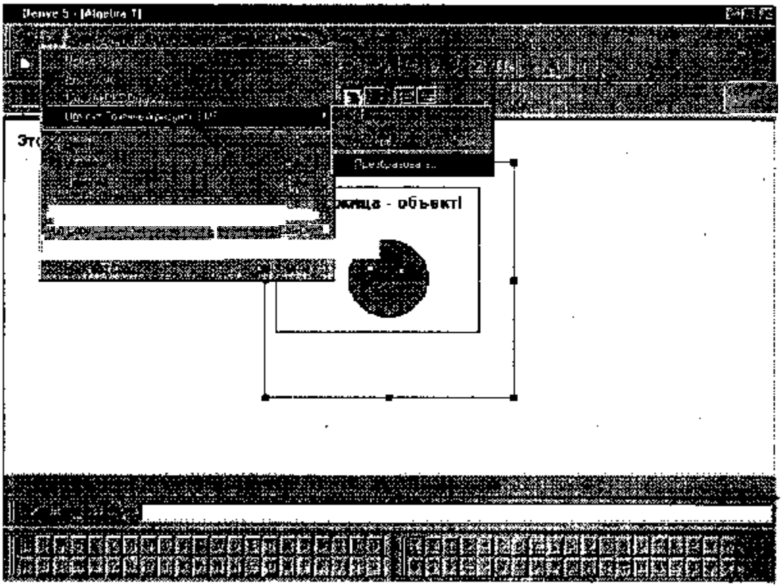


Рис. 8.16. Подготовка к преобразованию объекта в значок

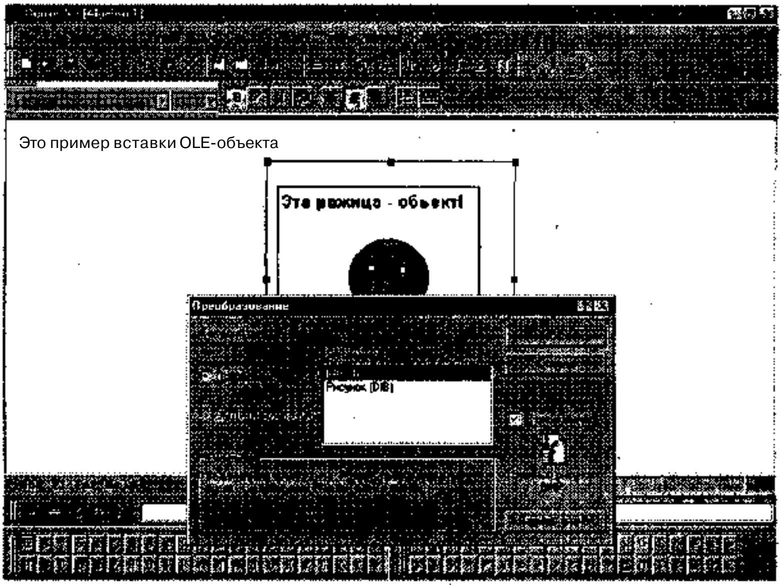


Рис. 8.17. Установка ОПЦИЙ преобразования объекта в значок

преобразовать уже имеющийся объект в значок. Для этого достаточно выделить объект и исполнить команду **Преобразовать,** доступную из подменю **Edit** (рис. 8.16).

После исполнения этой команды появится окно преобразования, показанное на рис. 8.17. Для представления объекта в виде значка надо активизиро-

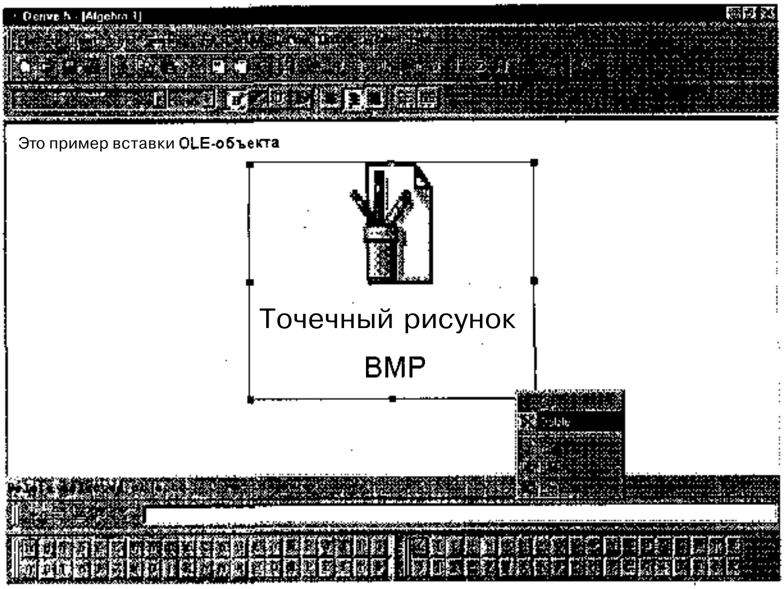


Рис. 8.18. Представление объекта в виде значка

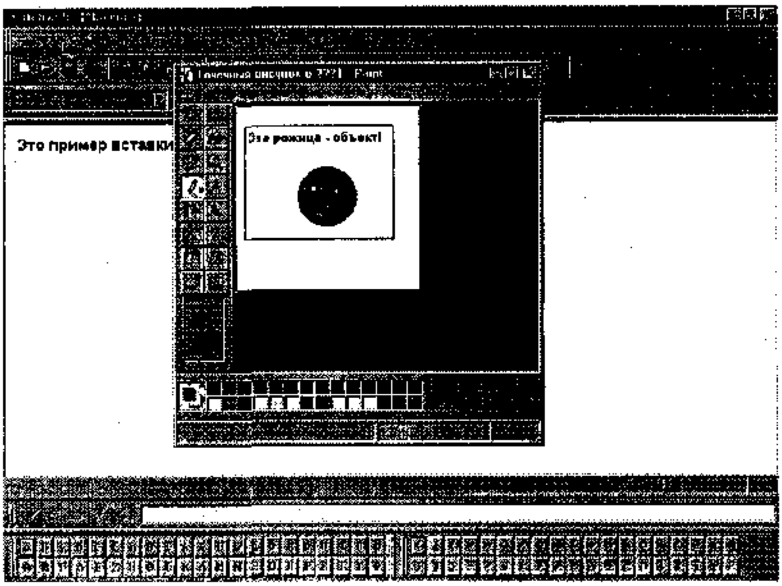


Рис. 8.19. Просмотр и редактирование объекта, представленного в документе значком

вать опцию **«В виде значка»**, установив в ее окошке мышью знак птички. Кнопка **Изменить значок** позволяет сменить значок, если представленный Е окне преобразования значок чем-то вас не устраивает.

Установив нужные опции, можно нажатием кнопки **OK** завершить преобразование. При этом объект, ранее существующий в форме конкретного рисунка, превращается в выбранный значок (рис. 8.18). Изображение значка можно также растягивать в разные стороны мышью.

Если нужно наблюдать объект, то достаточно щелкнуть дважды левой клавишей мыши, наведя предварительно ее курсор на значок. При этом запускается создавшее объект приложение и появляется его окно с созданным объектом (рис. 8.19). Характерно, что при этом данное окно уже имеет самостоятельный статус. В нем можно редактировать объект. По завершении его и выходе из приложения в документе Derive вновь появится значок, но уже привязанный автоматически к отредактированному объекту.

Какое из представлений объекта (прямое или в виде значка) лучше, это уже повод для размышлений пользователя. Для небольших рисунков предпочтительно прямое представление объектов, а для больших — в виде миниатюрного значка.

Вставка объекта из файла

Объект, например тот же рисунок, может быть представлен файлом. Для вставки такого объекта в окне вставки надо использовать опцию **«Создать из файла»** (рис. 8.20). Кроме того, можно задать связь объекта с файлом, установив опцию **«Связь».**

Используя кнопку **Обзор,** можно осуществить поиск файла, который представляет выбранный объект. По завершении этого нажатие клавиши OK вызывает появление объекта в документе Derive 5. Это показано на рис. 8.21,

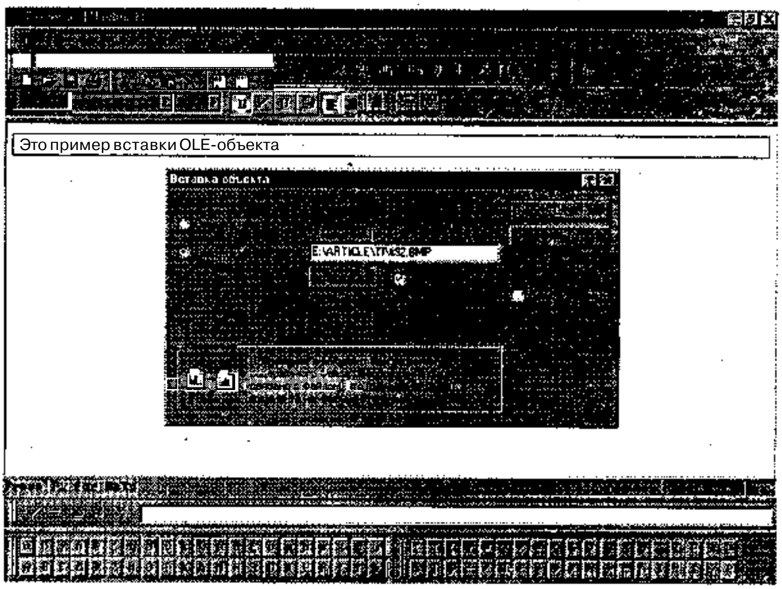


Рис. 8.20. Подготовка к вставке объекта из файла

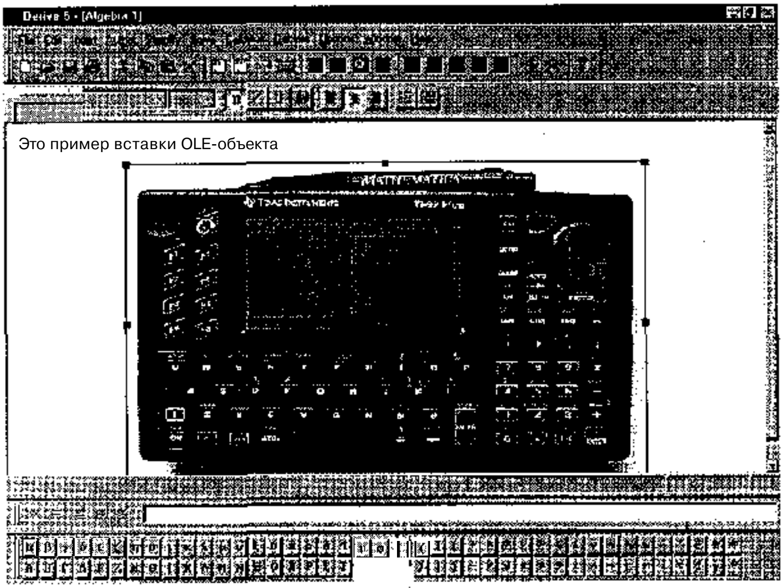


Рис. 8.21. Пример вставки объекта — рисунка из файла

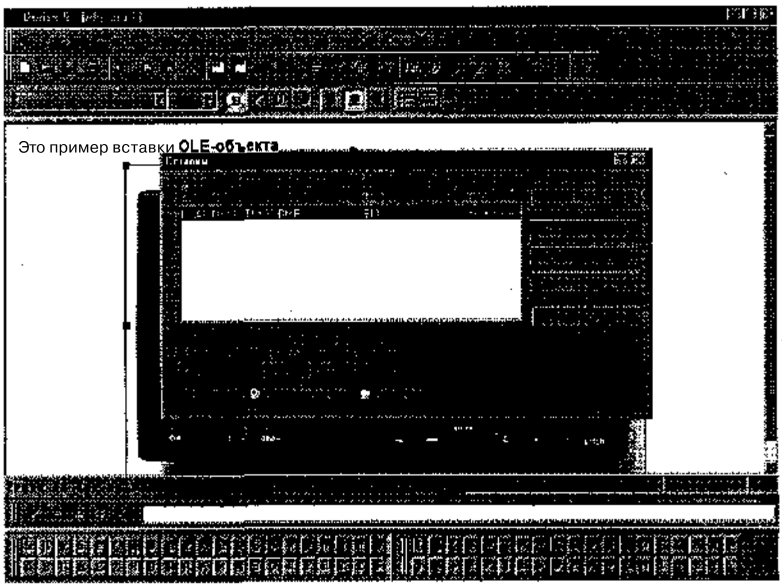


Рис. 8.22. Редактирование связи с объектом

где в качестве объекта выбрано изображение графического микрокалькулятора TI-92 Plus.

С подобной **вставкой** могут быть определенные проблемы из-за неполной совместимости форматов файлов, представляющих рисунки с форматами файлов, допустимыми для Derive. Это может ухудшать качество изображения (что и видно при внимательном просмотре рис. 8.21) или даже привести к невозможности представления объекта. В подобном случае целесообразно переписать объект в ином формате с помощью подходящего приложения, например графического редактора для рисунков.

Если установлена опция **«Связь»** (см. рис. 8.20), то существует возможность редактирования объектной связи. В этом случае в подменю **Edit** становится доступной команда **Links to OLE Objects...,** которая при ее исполнении выводит окно редактирования связи. Оно показано на рис. 8.22.

Возможности редактирования связей вполне очевидны из русскоязычных надписей в этом окне. Напоминаем, что появление таких надписей указывает на то, что данное окно принадлежит не системе Derive, а русифицированной операционной системе, установленной на ПК.

В целом важно отметить, что использование механизма объектной связи в системе Derive 5 позволяет (наряду с другими, отмеченными выше возможностями) создавать в ее среде вполне полноценные ноутбуки, содержащие такие объекты, как произвольные красочные рисунки и тексты.

Установка опций Derive 5

Позиция **Options** меню подверглась заметным изменениям. Первые три ее команды **(Display, Printing и Sturtup...)** остались прежними (как в Derive 4.* под Windows). Зато другая часть опций новая для этой позиции. В нее перенесена из позиции **Edit** опция перенумерации строк документа.

Однако главным изменением стало введение опций скрытия ряда важнейших объектов документа:

Hide Labels — скрытие номеров строк документа;

Hide Plots — скрытие графических блоков;

Hide Text — скрытие текстовых блоков;

Hide OLE Objects — скрытие вставленных в документ OLE-объектов.

Введение этих новых опций позволяет строить интересные документы, облегчающие построение лекционных курсов. Например, вы можете скрыть пояснения математических понятий и графиков, предлагая учащемуся обдумать оставшиеся в документе математические выкладки. Как недостаток такого подхода надо указать, что опции действуют не выборочно, а глобально (применительно ко всему документу).

Вкладки окон открытых документов

В позиции **Windows** меню наряду с типичными вариантами расположения окон имеется новый вариант — в виде легко переключаемых вкладок. Для этого введена опция **Display Tab.** Вид окна Derive 5 с тремя открытыми окнами и тремя вкладками показан на рис. 8.23.

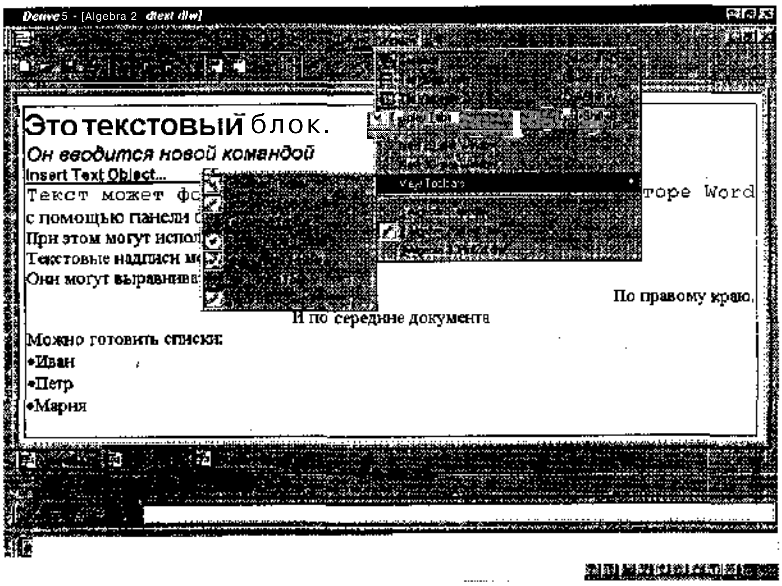


Рис. 8.23. Вид окна Derive с вкладками окон документов

Подобный вид представления окон введен в новые версии табличных процессоров Exel, и в Derive 5 заимствована идея такого представления окон. Оно очень удобно при работе с рядом документов и перенесении объектов из одного окна в другое.

Новые возможности графики Derive 5

Построение **3D-фигур** с функциональной окраской и параметрическим заданием

Трехмерная графика Derive 5 существенно продвинута по сравнению с предшествующими версиями и обеспечивает все основные возможности такой графики, применяемые в современных системах компьютерной математики. Как уже отмечалось, в эту версию впервые введена возможность функциональной окраски трехмерных фигур и поверхностей.

Интересной является возможность построения 3D-графиков при параметрическом задании их списком из трех функций. Каждая из них определяет координату x, y и z поверхности как функцию от длины радиус вектора и углов s и t (см. пример на рис. 8.24).

Как видно из приведенного рисунка, он представляет собой отрезок толстой спирали в пространстве. По наглядности этот график ничуть не уступает подобным графикам, которые строят более мощные системы компьютерной математики, например Mathcad, Maple и Mathematica.

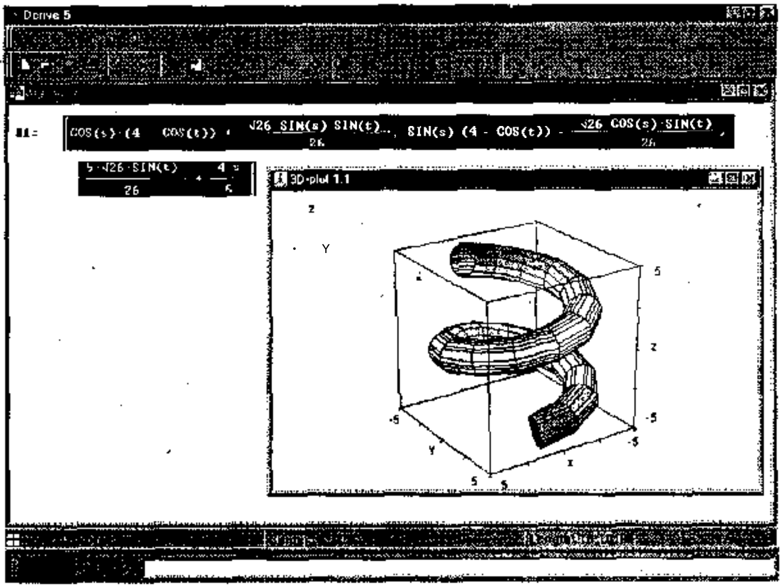


Рис. 8.24. Пример построения **3D-фигуры,** заданной параметрически

Построение пересекающихся в пространстве фигур

*

Еще одна интересная возможность графики Derive 5 — построение пересекающихся в пространстве поверхностей или трехмерных фигур. Для этого достаточно задать список выражений для таких фигур в виде функций двух

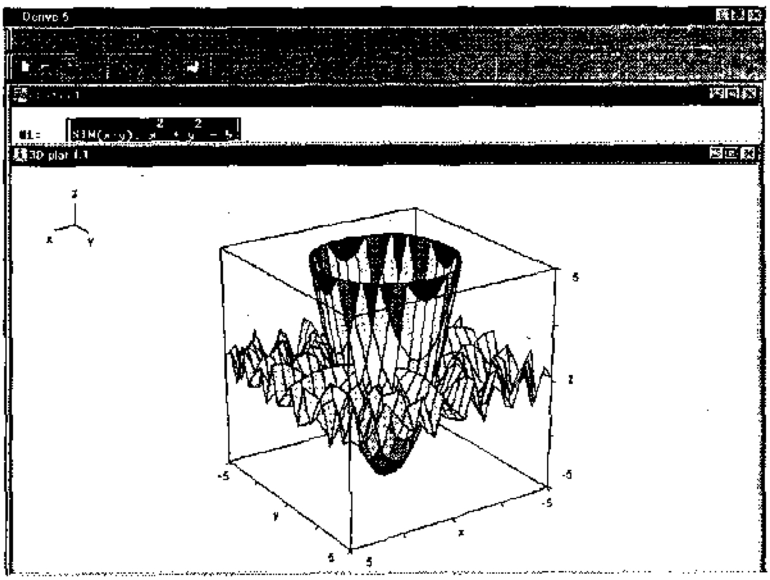


Рис. 8.25. Построение двух пересекающихся в пространстве поверхностей

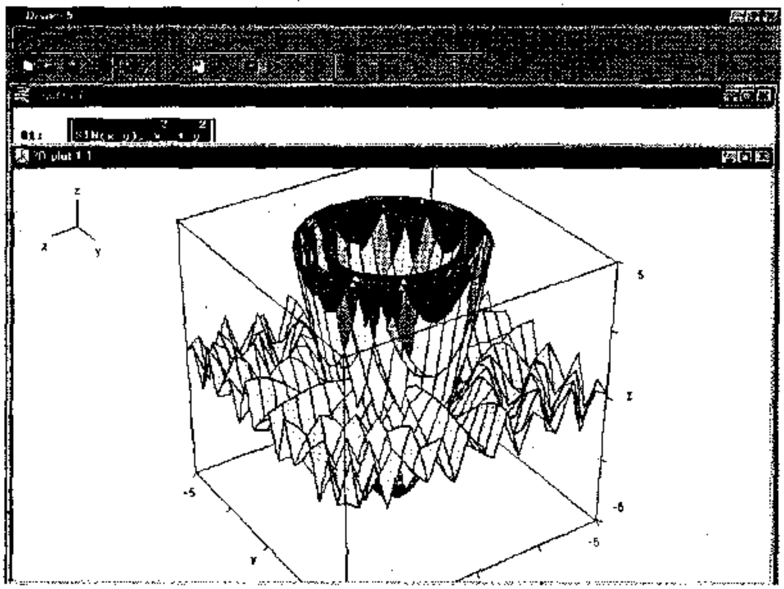


Рис. 8.26. Построение 4 поверхностей в пространстве

переменных x и y, задающих координаты точек z(x, y) каждой из фигур. Рис. 8.25 показывает пример такого построения для двух поверхностей.

В данном случае построены графики поверхности sin(x*y) и смещенной вниз пространственной параболы ($x^2 + y^2$). Нетрудно заметить, что Derive 5 неплохо справляется с построением линий пересечения поверхностей в пространстве и эффективно использует алгоритм удаления невидимых частей поверхностей. Это делает графики (особенно при использовании функциональной окраски) весьма наглядными и полезными при проведении уроков по стереометрии.

Интересно, что эту возможность можно сочетать с последовательным построением трехмерных объектов на одном рисунке. Например, можно построить график поверхности sin(x*y), а затем три пространственные параболы со сдвигом по оси z, равным -5, -3 и без сдвига. Конечный результат такого построения представлен на рис. 8.26.

Можно продолжить эксперименты с построением ряда фигур в пространстве. Рис. 8.27 показывает построение 5 поверхностей в пространстве. Обратите внимание на то, что в ограниченную область пространства попадают только части некоторых фигур.

2D-графика с окраской по неравенствам

Еще один очень любопытный и полезный вид двумерной графики есть у системы Derive — это графики с окраской отдельных их сегментов, задаваемых с помощью неравенств. Пример такого графика приведен на рис. 8.28.

Как нетрудно заметить, в выражении для построения такого графика заданы синусоиды с амплитудами, равными 0.5 и 1. При этом ограничения за-

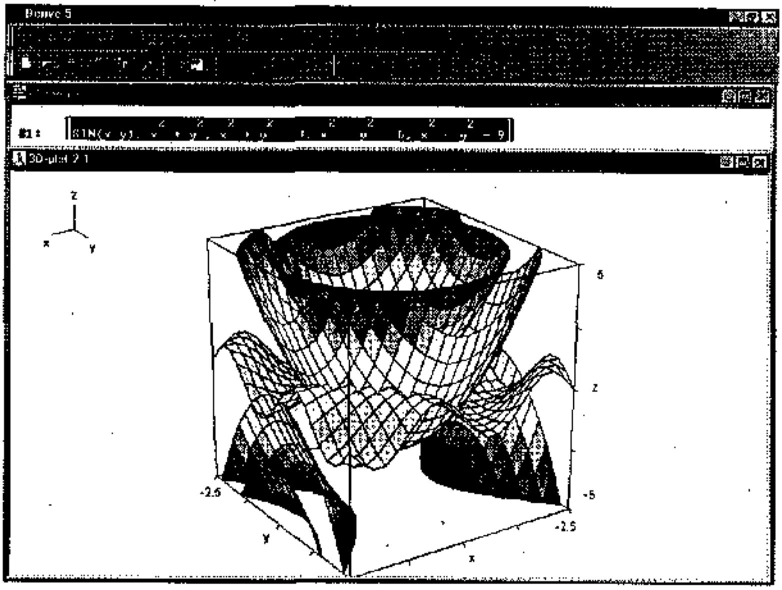


Рис. 8.27. Построение 5 поверхностей в пространстве

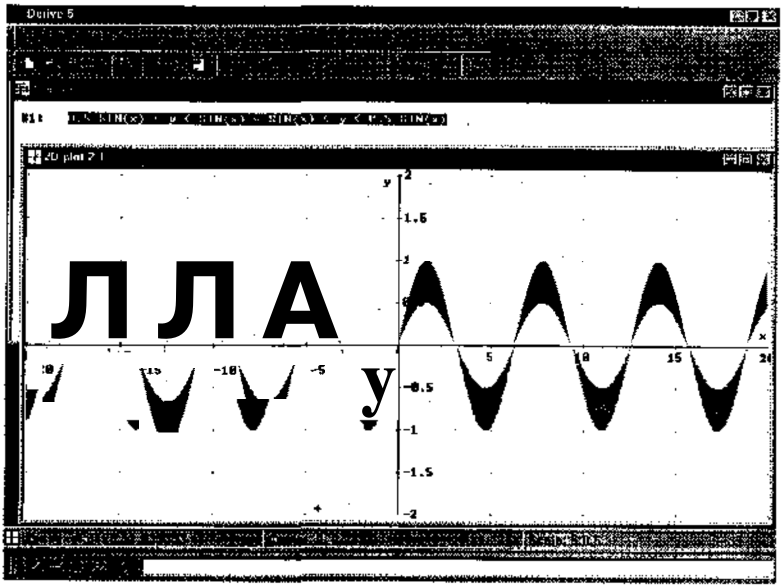


Рис. 8.28. Построение площадей, ограниченных синусоидами с амплитудой, равной **0.5** и 1

писаны как положительной, так и отрицательной полуволн синусоиды. Область между этими двумя синусоидами строится окрашенной.

Следующий рисунок показывает построение графика функции sin(x)/x с закраской областей, ограниченных графиком функции и осью абсцисс x

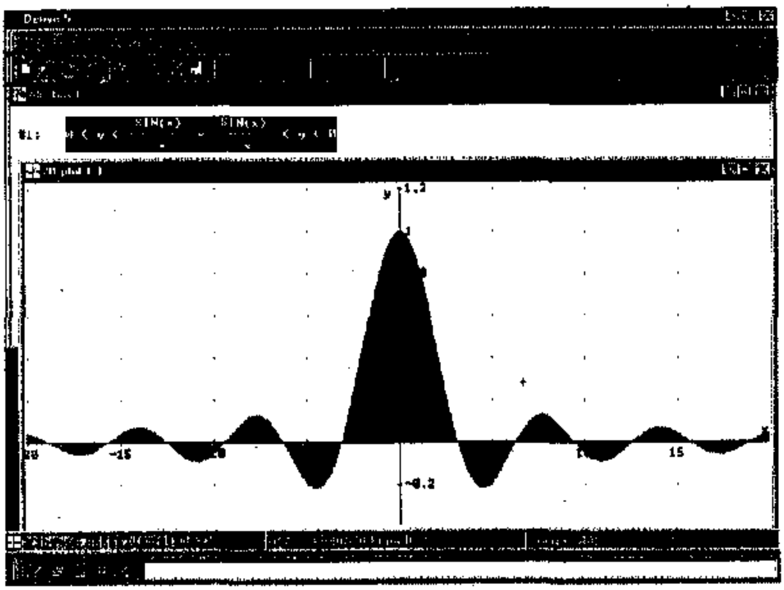


Рис. 8.29. График функции sin(x)/x с разной окраской положительных и отрицательных полуволн

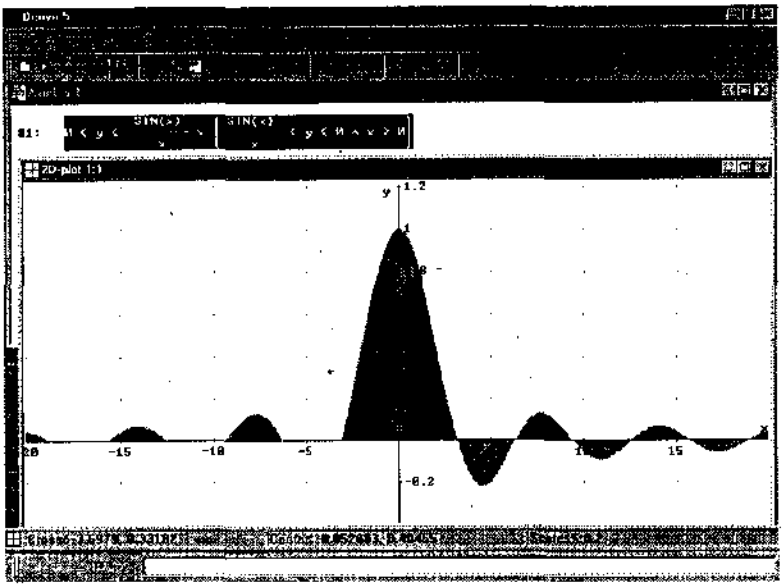


Рис. 8.30. График функции sin(x)/x при комбинированном задании условий окраски

(рис. 8.29). При этом окраска положительных и отрицательных полуволн выполняется разным цветом.

Следующий пример (рис. 8.30) показывает, что помимо ограничений на значения **y** можно ввести и ограничение на изменение x. В данном случае для значений $\sin(x)/x<y$ задано еще одно условие — $x>0$. Это означает, что будут закрашиваться только отрицательные полуволны зависимости $\sin(x)/x$ для $x>0$.

Приведенные примеры позволят читателю опробовать свои силы в построении подобных графиков.

Иллюстрация численного интегрирования

Одной из полезных возможностей графики описываемого типа является графическая иллюстрация численного интегрирования. В качестве примера рассмотрим вычисление простого интеграла.

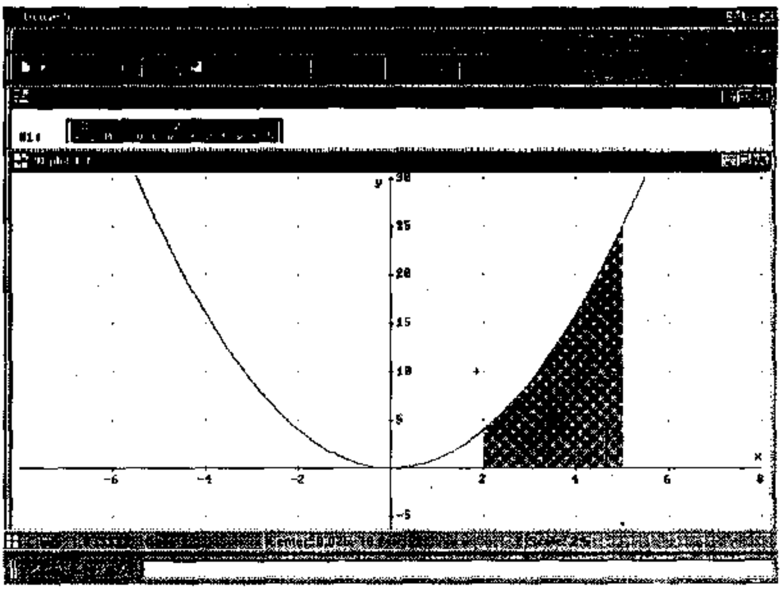


Рис. 8.31. Графическая иллюстрация численного интегрирования

Этот интеграл можно представить как площадь параболы x^2, ограниченную осью x, кривой параболы и вертикалями с абсциссами 2 и 5. Рис. 8.31 показывает, как можно проиллюстрировать это положение. На нем показано построение параболы и площади, представляющей значение интеграла.

Внимание

Читателю рекомендуется самостоятельно построить иллюстрации к численному интегрированию для других подынтегральных функций, да так, чтобы площади сверху и снизу оси абсцисс различались цветом.

Построение «шахматной доски*

Еще один пример на построение «шахматной доски» с частично закрашенными квадратами представлен на рис. 8.32.

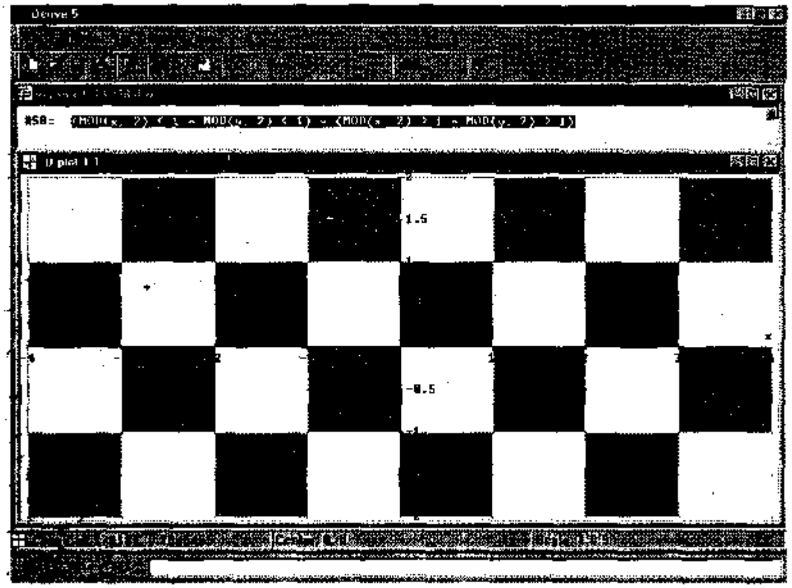


Рис. 8.32. Построение «шахматной доски»

Графика с закраской до неравенствам открывает обширные возможности в создании сложных фигур типа палитр или наблюдаемых в калейдоскоп узоров. Очень интересные примеры такой графики даны в файле patts2d.dfw, размещеаном в пакете лучших программ пользователей. Его можно скачать с Интернет-сайта фирмы Texas Instruments (см. урок 1).

Управление средствами графики

Управление средствами 2D-графики

Управление графическими окнами в Derive 5 организовано, как и в предшествующей версии, с помощью меню и панели инструментов. Они в графических окнах иные, чем у окна документа. В дополнение к этому возможно управление с помощью контекстно зависимого меню правой клавиши мыши.

Поскольку управление графическими окнами для Derive 4 под Windows было подробно описано, ограничимся указаниями на те новые средства, которые появились в Derive 5.

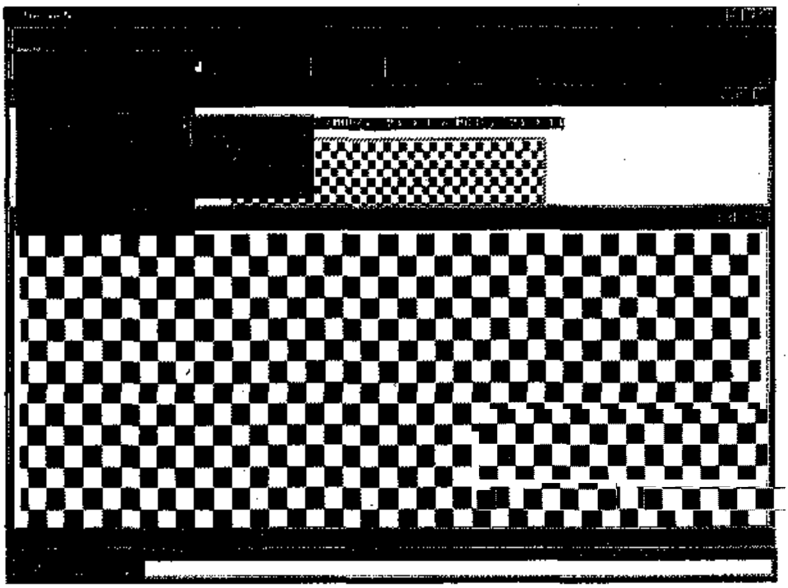


Рис. 8.33. Окно Derive 5 с окном графики и открытым подменю File

В подменю File (рис. **8.33,** где это подменю открыто) можно найти ряд новых команд:

Embed — вставка графика в документ;

Update — перестройка графика в документе, если использовалась команда Embed;

Export — экспорт графика в файл заданного формата.

Наиболее важной здесь является команда Embed, вставляющая график, имеющийся в графическом окне, в документ. Если команда Embed уже применялась и в документе есть вставленный график (он виден на рис. 8.32 над окном графики), то при модификации графика ввод его изменений в окно документа производится командой Update (если исполнить в этом случае команду Embed, то новый график разместится под уже имеющимся).

Весьма полезна также новая команда Export, позволяющая записывать содержимое графического окна в файлы различного формата (они перечислены в подменю **File-Export).** Важной также является группа команд, относящаяся к печати графика принтером. Она реализует типовые средства печати, присущие приложениям под Windows **95/98/NT,** но несколько упрощенные. Так, команда Print Setup **выводит** окно установки только полей печатаемого рисунка (рис. 8:34).

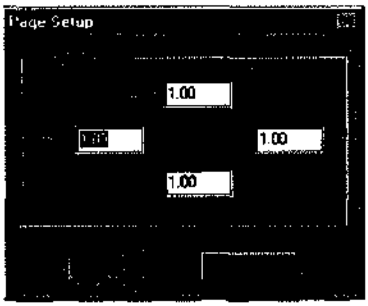


Рис. 8.34. Окно установки полей печатаемого рисунка

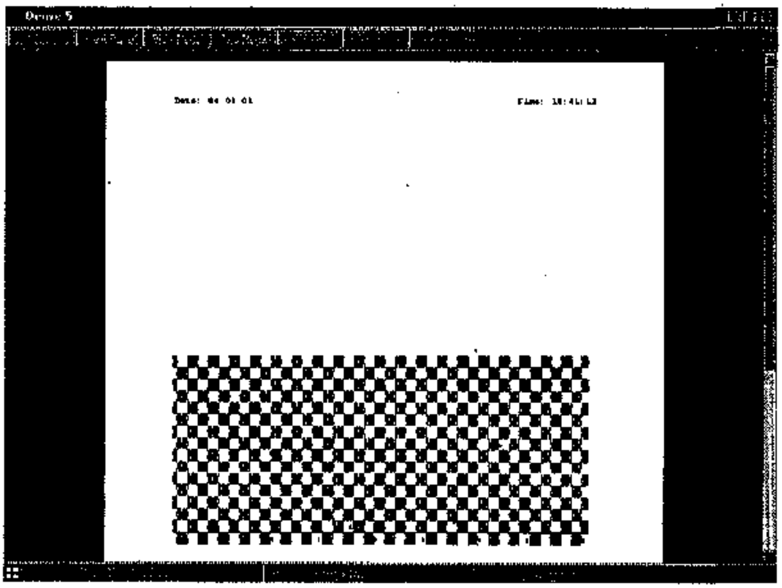


Рис. 8.35. Окно предварительного просмотра графика перед печатью

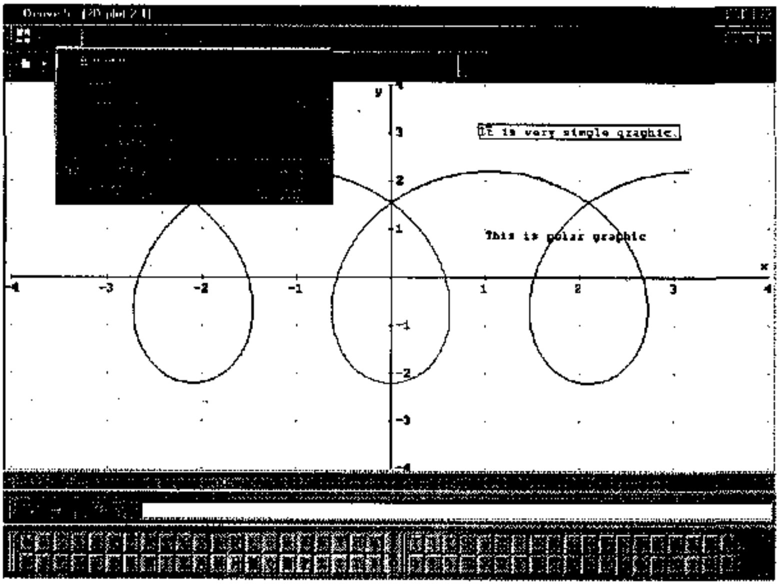


Рис. 8.36. Окно Derive 5 с открытым подменю Edit

Команда Print Preview выводит окно просмотра графика перед печатью (рис. 8.35). Органы управления этого окно малочисленны и вполне очевидны. Прямо из окна (или командой Print из подменю File) можно задать распечатку графика. Точнее говоря, команда Print выводит окно того принтера, который используется компьютером для печати. Кнопка **Close** закрывает окно предварительного просмотра.

Позиция **Edit** меню окна графики претерпела несущественные изменения — предусмотрены новые команды стирания всех графиков и всех аннотаций (рис. 8.36). Новыми возможностями Derive 5 стали: выделение аннотации прямоугольником и их плавное перемещение мышью в любое место, а также редактирование аннотации.

Команда **Create Annotation** (создание аннотации) из подменю **Edit** удалена. Она перешла в подменю **Insert** меню графического окна. В это подменю переведена и отдельная позиция **Plot** (построение графика), бывшая в прежней версии Derive 4.* под Windows. Эти изменения логичны, но вряд ли принципиальны.

Мало изменилось и меню установок **Set.** К бывшим там позициям добавилась новая — **Coordinate System...** (установка координатных систем — прямоугольной или полярной). Раньше она была в подменю **Options.** Последнее в Derive 5, за исключением отмеченного перемещения, практически не **изменилось.** Добавилась только обсужденная ранее опция **Simplify Before Ploting** (упрощение выражения перед построением графика). И наконец, в подменю **Window** добавлена команда **Display Tabs** задания окон документов в виде вкладок.

Управление средствами **3D-графики**

Заметные улучшения введены в управление средствами трехмерной графики. Как уже отмечалось, панель инструментов окна **3D-графики** пополнилась новыми кнопками для поворота объектов вокруг осей x и y, приближения и удаления объекта и анимации путем вращения объекта вокруг оси z.

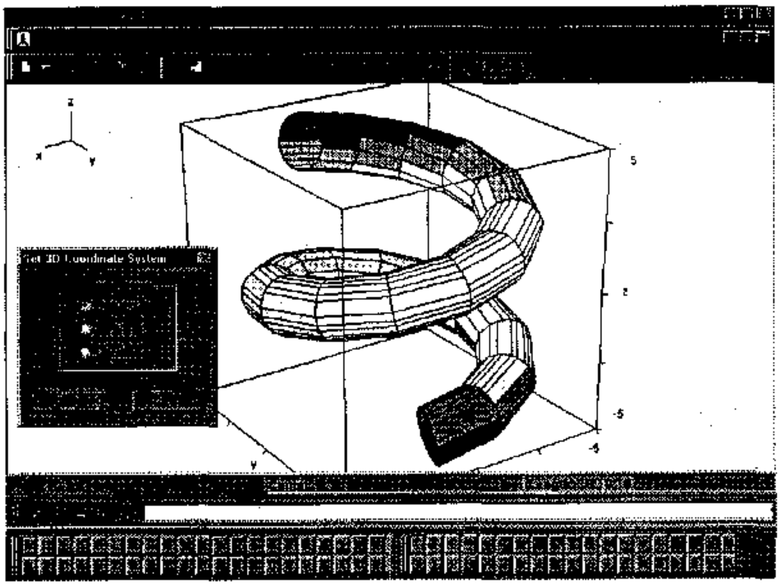


Рис. 8.37. Окно 3О-графики с окном выбора координатной системы

7*

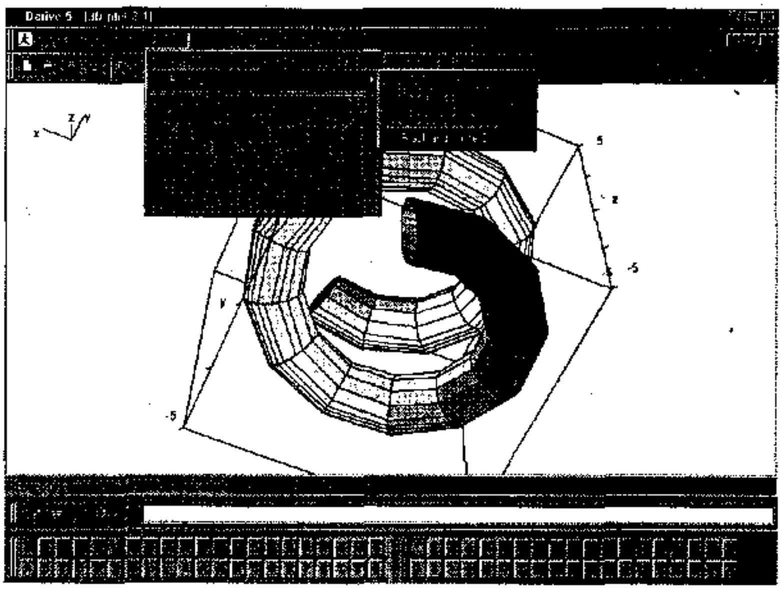


Рис. 8.38. Окно **3D-графики** с открытым подменю опций

Из новаций в меню окон **3D-графики** надо отметить расширение установок в позиции **Set.** Наиболее важным и новым тут стала возможность выбора одной из трех координатных систем (рис. 8.37). Это следующие системы: **Rectangular** (прямоугольная), **Spherical** (сферическая) и **Cylindrical** (цилиндрическая). Таким образом, есть возможность перестройки графика в любой из этих координатных систем (окно выбора их показано слева от графика на рис. 8.37).

Заметной модификации подверглось меню опций **3D-графики Options.** Набор опций в нем представлен на рис. 8.38. Новыми опциями являются опции анимации (вращения) **3D-объекта,** упрощения до построения графика и аппроксимации (вычисления в численном виде) до построения графика.

Меню **Windows** и **Help** модификации не подвергались. В целом графика Derive 5 представляет пользователю (как педагогу, так и учащемуся) поистине неограниченные возможности в графической визуализации сложных математических понятий и образов. Впору написать о ней отдельную книгу, но мы вынуждены ограничиться представленным выше материалом.

Новые возможности в вычислениях

Улучшение функции SOLVE

Ряд новых возможностей в Derive 5 введен в средства вычислений. Некоторые из них представлены на рис. 8.39.

К числу таких возможностей относится применение в реализации функции **SOLVE** нового базового алгоритма **Гробнера,** позволяющего разыскивать

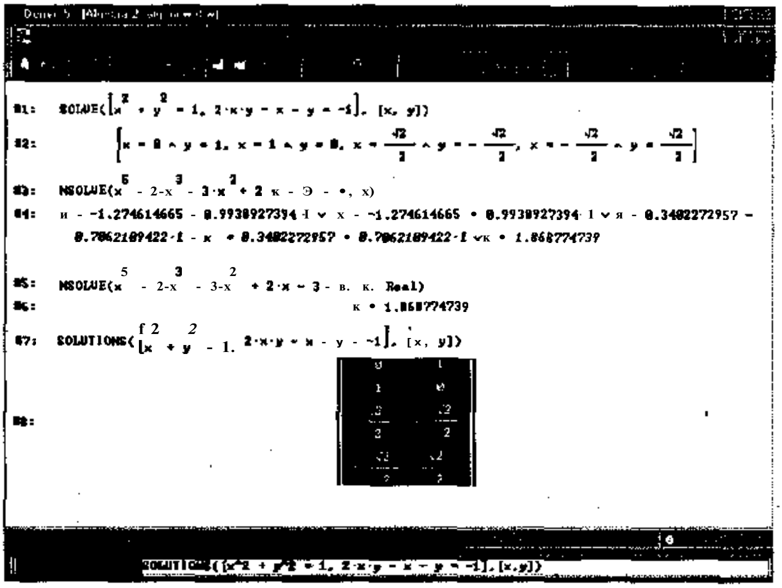


Рис. 8.39. Демонстрация новых возможностей Derive 5 в вычислениях

точные решения систем нелинейных полиномиальных уравнений, в том числе с результатами в виде логических условий (см. пример в строках 1 и 2 на рис. 8.39).

Новая функция NSOLVE

При переходе к численным решениям полезна новая функция **NSOLVE.** Ее применение показано в строках **3—6** рис. 8.39. Обратите внимание на применение во втором случае (строка 5) опции **Real** в данной функции, позволившей отбросить комплексные корни. Эта функция может использоваться и для решения систем нелинейных уравнений и неравенств в численной форме.

Новые функции SOLUTIONS и NSOLUTIONS

Формат решений функции **SOLVE** имеет особый вид. Он удобен для обзора решений, но неудобен для их дальнейшего использования. От этого недостатка избавлена новая функция **SOLUTIONS,** введенная в Derive 5. Она возвращает решение в виде вектора или матрицы (см. пример в строках 7 и 8 рис, 8.39). При решении уравнений и неравенств (и их систем) в численном виде может использоваться еще одна новая функция **NSOLUTIONS.**

Символьные операции с логическими выражениями

С логическими выражениями в Derive 5 могут выполняться различные символьные операции компьютерной алгебры. На **рис.** 8.40 приведены примеры таких операций с использованием функций **EXPAND, TERM, FACTOR,**

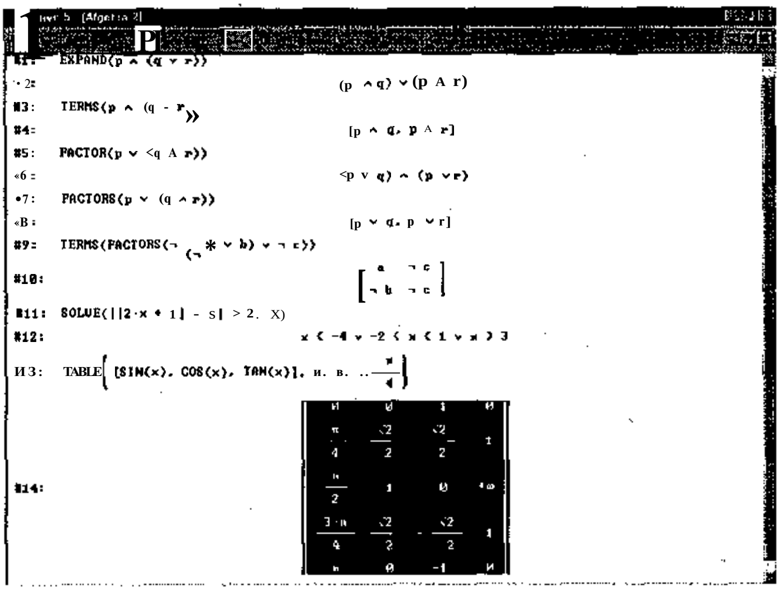


Рис. 8.40. Операции с логическими выражениями и созданием таблицы

FACTORS и **TERMS.** Все приведенные на рис. 8.40 результаты получены исполнением команды **Simplify Basic.**

Функции **LHS** и **RHS** теперь также расширены и могут выделять левую и правую часть выражений с бинарными операторами.

Новая функция табуляции TABLE

Теперь для создания векторов и матриц в форме таблиц в Derive 5 появилась новая функция

TABLE(u, k, п) — создает таблицу значений и (одиночное выражение или вектор с выражениями) при изменении к от 1 до п;

TABLE(u, к, п1,п2) — создает таблицу значений и при изменении к от **n1** до п2;

TABLE(u, к, n1,n2,d) — создает таблицу значений и при изменении к от nl до п2 с шагом d.

Пример применения этой функции дан на рис. 8.40.

Расширенные возможности декларации переменных

Derive 5 имеет расширенные возможности декларации переменных. При исполнении команды **Declare — Variable Domain...** появляется окно декларации переменных, представленное на рис. 8.41.

Помимо применяемых в прежних версиях типов переменных **Integer** (целочисленная), **Real** (вещественная) и **Complex** (комплексная), введено три новых типа: **Vector** (вектор), **Set** (множество) и **Logical** (логического типа). Расширены также возможности задания интервалов допустимого изменения

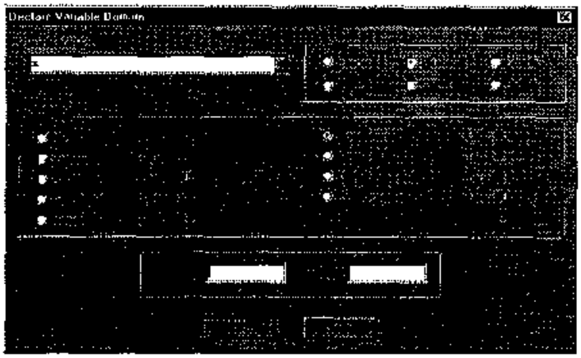


Рис. 8.41. Окно декларации типов переменных Derive 5

переменных типа **Integer** и **Real.** Теперь их можно задавать в открытом и закрытом интервалах.

Внимание

*Многие задачи имеют решение только при задании переменным определенных свойств, например, вычисление логарифма возможно для действительных положительных чисел. Поэтому расширение возможностей декларации свойств переменных означает **расширение** класса решаемых задач.*

Чему мы научились:

- Устанавливать и запускать Derive 5 под Windows.
- Оценивать новые возможности Derive 5.
- Использовать расширенные возможности редактирования.
- Применять операции вставки.
- Устанавливать опции Derive 5.
- Работать с вкладками окон открытых документов.
- Использовать новые возможности графики Derive 5.
- Использовать новые возможности Derive 5 в вычислениях.

Урок 9. Типовые средства программирования

- Алфавит системы и комментарии
- Основные типы данных
- Переменные и функции
- Арифметические и логические операторы
- Функция IF для создания условных выражений
- Функция для организации итераций
- Практические приемы программирования
- Функции выделения различных частей отношения
- Расширенные средства программирования в Derive 5
- Заключительные замечания по программированию

Алфавит системы и комментарии

Понятие о входном языке Derive

Derive создавалась как система, не требующая программирования при решении многих математических задач. Однако это не означает, что программирование при работе с этой системой не нужно вообще. Какой бы мощной ни была математическая система, она не может решать любые задачи. Хотя бы потому, что всегда возникают все новые и новые задачи, о которых создатели системы попросту 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, на котором написана сама система (muLISP), называется входным языком.

Derive (см. предшествующие уроки) позволяет реализовать многие массовые виды вычислений в командном (непосредственном) режиме вычислений. 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 в язык входят 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. Языки-компиляторы воспринимают всю программу сразу и создают быстро исполняемые коды в виде EXE- или COM-файлов. Поэтому скорость выполнения программ у них выше, но нужен этап компиляции программы.

Интерпретирующий режим работы общепринят у подавляющего большинства математических систем, лишь Mathematica 2/3/4 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 является проблемно-ориентированным на решение математических задач языком функционального программирования сверхвысокого уровня. Derive имеет множество встроенных функций (например, для вычисления пределов, производных, интегралов и т. д.), которые, при реализации их на обычных языках программирования, потребовали бы многих месяцев напряженной работы программистов.

Программы для системы Derive хранятся на магнитных дисках в виде программных файлов. Они могут быть двух видов. Файлы с расширением .mth после загрузки требуют запуска. Демонстрационные файлы с расширением .dmo имеют автозапуск после загрузки — каждая строка их 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 ASCII. Эти файлы могут содержать лишь допустимые для входного языка Derive символы и слова, в совокупности образующие алфавит этого языка.

Алфавит Derive

В алфавит входного языка системы Derive входят латинские буквы от a до z (и от A 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 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	(у)
Alt-Q	**SQRT** — квадратный корень	

Для отличия от обычных знаков спецзнаки e и i на экране дисплея выделяются символом птички ^над ними. Этот знак указывает на особый статус этих букв. Буква в скобках указывает на то, как отображается этот знак при использовании драйвера русификации системы в системах Derive под MS-DOS.

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 — нечто 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. Таким образом, можно комментировать ход вычислений по каждой строке, не засоряя верхнюю часть экрана большим числом пояснений. Этот прием использован в демонстрационных программах (файлах) Derive, и вы можете применить его в своих примерах. Для ввода подобных комментариев годится любой текстовый редактор, поддерживающий формат ASCII.

Основные типы данных

Константы и числа

Константы, это имеющие имя объекты, значения которых неизменно и задается перед началом работы с системой автоматически. В алфавит Derive входят четыре числовые константы:

pi или *п* — число «пи» 3.14159265 (вводится как **Alt-P),**
deg или ° — константа (pi/180) для перевода углов из градусов в радианы,
inf или оо — положительная машинная бесконечность;
euler_gamma — константа Эйлера 0.5772156649.

К логическим константам относятся **true** (истинно) и **false** (не истинно). Числа относятся к константам, именем которых являются их значения. 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 вида **a+b*i,** где i — мнимая единица (корень квадратный из -1), используется константа #i, отображаемая дисплеем как i с птичкой сверху. Константа #e (на экране дисплея ê с птичкой сверху) является основанием натурального логарифма.

Константы вместе с функциями могут входить в состав арифметических выражений, например:

```
5 * sin(2 * pi * x+ phi)
```

Здесь 5, 2 и pi — константы, sin — наименование функции с аргументом в круглых скобках и phi — имя переменной (см. следующий разд.).

Векторы и матрицы

Наряду с отмеченными выше простыми типами данных (целые, вещественные и комплексные числа) Derive имеет более сложные типы данных — это векторы и матрицы. Вектор — это просто последовательность чисел

X1 X2 X3 ... Xi ... Xn

или символьных выражений. Она образует одну строку (или один столбец) данных.

Матрица является совокупностью m векторов одинакового размера. Далее рассматриваются матрицы с размерностью **m*n**, где m — число строк матрицы, **а n** — число столбцов. Такая матрица может быть представлена в виде

A1,1	A1,2	A1,3	...	A1,n
A2,1	A2,2	A2,3	...	A2,n
. . .	. . .	. . .	Aj,i	...
Am,1	Am,2	Am,3	...	Am,n

Каждый элемент вектора можно представить в виде **Vj**, а матрицы (например, А) в виде Aj,i, где **j** и i — индексы, определяющие положение элемента в векторе или матрице. Матричные операторы и функции будут подробно описаны в следующем уроке.

Множества

Множества относятся к не вполне определенным математическим понятиям, не имеющим точного определения. Так, может быть множество предметов, мыслей или людей. Формально в Derive множества задаются в виде списков объектов (чисел, констант, переменных и т. д.), заключенных в фигурные скобки. Например, {1,7,3,5,2} — множество из пяти целых чисел.

Переменные и функции

Переменные и их определение

Переменные в Derive могут быть необъявленными и 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 (:=):

Имя_переменной := Выражение

Имя переменной может иметь одну букву или состоять из ряда букв и цифр, но начинаться 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 или вектор или матрицу значений. Как отмечалось, при загрузке драйвера кириллицы возможно применение в идентификаторах букв русского алфавита. Не стоит, однако, широко практиковать это — 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 в любом месте программы и их можно использовать в любом месте.

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).** Большая часть греческих букв является идентификаторами предопределенных переменных. Но некоторые, например константа π или обозначение гамма-функции Г, зарезервированы под специальные цели.

Однако отмеченным образом нельзя поступать 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 EXP(a b c), т.е. по сути будет **exp(a*b*c).** Поэтому переменные с составным именем должны предварительно объявляться (см. разд. 1.7).

В некоторых символьных операциях вводится понятие приоритета переменных. Например, если выражение содержит несколько переменных **(x,y,z** и т. д.), то 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.

Встроенные функции

Важным понятием систем компьютерной математики является понятие функции. В Derive оно совпадает с таким понятием в математике. Функция — это поименованная операция преобразования некоторых входных данных в выходные данные, которые этой функцией возвращаются. Итак, важным признаком функции является возврат ею преобразованных значений аргументов (параметров) функции.

Все функции Derive возвращают единственное значение. К примеру, функция **SIN(1.0)** имеет параметров вещественное число 1.0 и возвращает также вещественное число, развное значению синуса от единицы. Однако если задать SIN(1) с аргументом — точной единицей, то будет возвращено значение

SIN(1) без его перевода в вещественное значение. Это связано с тем, что SIN(1) является точным значением, тогда как **SIN(1.0)** приближенным.

Derive имеет множество встроенных в ядро функций. Такие функции мы будем называть встроенными. Ядро Derive оптимизировано для быстрого выполнения операций с такими функциями. Ввиду обилия встроенных функций знакомству с ними посвящен отдельный (следующий) урок.

Функции пользователя **и** их декларация

В программировании функция это некоторая вычислительная операция над заданными параметрами, возвращающая, 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 имеется возможность задания новых функций в виде функций пользователя. При этом 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 только 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), то переменная x будет иметь значение 1, а y значение 2 только в выражении sin(x) + cos(y), представляющим собой тело функции. В самой же программе эти переменные могут обозначать все что угодно, например координаты точки, и иметь произвольные иные 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.

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 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 дан пример, наглядно проясняющий разный статус 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.

Арифметические и логические операторы

Форматы чисел

При арифметических операциях Derive может работать как с действительными, так и комплексными числами. О действительных числах — константах уже говорилось. Отметим теперь комплексные числа.

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

z = Re(z) + i*Im(z),
w = Re(w) + i*Im(w),

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

Арифметические операторы

Для выполнения арифметических операций используются операторы в виде обычных математических знаков, принятых для таких операций:

-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 более высокий приоритет, но над ними возвышаются логические операции и вычисление функций. При задании комплексных чисел мнимая единица вводится как #i.

Примеры арифметических операций с комплексными числами даны ниже:

```
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: $2\ W$

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

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

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

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

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

12: $\mathrm{IM} \left(\dfrac{a\ c}{c^2 + d^2} + \dfrac{b\ d}{c^2 + d^2} + \#i \left(\dfrac{b\ c}{c^2 + d^2} \quad \dfrac{a\ d}{c^2 + d^2} \right) \right)$

13: $\dfrac{b\ c}{c^2 + d^2} + \dfrac{a\ d}{c^2 + d^2}$

14: PHASE $(3 + 2\ \#i)$

15: $\dfrac{31661}{53845}$

16: CONJ $(3 + 2\ \#i)$

17: $3 - 2\ \#i$

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

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

Операторы отношений

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

u = v — равенство выражений;
u /= v — неравенство выражений;
u < v — выражение и меньше v;
и > v — выражение и больше v;
и <= v — выражение и меньше ИЛИ равно v;
и >= v — выражение и больше ИЛИ равно v.

Указанные выше конструкции дают результат true (ИСТИННО) или false (ЛОЖНО), который трактуется как логический. Объекты true и false в Derive заданы как логические константы.

Логические операторы

Derive имеет ряд логических операторов. Если У, У1 и У2 — условия, то эти операторы используются в виде:

NOT У — логическое отрицание;
У1 AND У2 — логическое сложение — операция «И»;
У1 OR У2 — логическое умножение — операция *ИЛИ»;
У1 XOR У2 — логическая операция «Исключающая ИЛИ»;
У1 IMP У2 — логическая операция «Импликация».

Действие этих операторов легко понять, считая, что true дает логическую единицу 1, а false — логический ноль 0. Тогда NOT дает 1, если У=0, и 0, если У=1. Таблица истинности логических бинарных операторов имеет вид:

У1	У2	AND	IMP	OR	XOR	
0	0	0	1	0	1	0 — FALSE (ЛОЖНО)
1	0	1	0	0	1	
0	1	1	0	0	1	1 — TRUE(ИСТИННО)
1	1	1	0	1	0	

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. Для изменения приоритета логических операций можно использовать круглые скобки.

Логические операторы могут иметь ряд эквивалентных форм, известных из курса Булевой алгебры. Над целыми операндами операторы **NOT**, AND, OR, **XOR** и **IMP** выполняют побитовые логические операции. Операции выполняются для «полубесконечного» дважды дополнительного представления целых. Например, 3 OR 5 упрощается к 7, а NOT 5 упрощается к -6.

Операторы для множеств

Для множеств определены следующие операторы:

s\` — дополнение для s (постфиксный оператор). Пример: s\`\` дает s.

s *n* t — пересечение s и t (вводится комбинацией клавиш **Ctrl-N** или вводом оператора INTERSECTION). Пример: {1,3,5,7,9}п{2,3,5,8,13} дает {3,5}.

s ∪ t — объединение s и t (вводится комбинацией клавиш **Ctrl+U** или вводом оператора UNJON). Пример: **{1,3,5,7,9}∪{2,3,5,8,13}** дает {1,2,3,5,7,8,9,13}.

s \ t — разность множеств s **и** t. Пример: **{1,...,10}\\{2,3,5,7}** дает **{1,4,6,8,9,10}**.

Операторы в виде специальных знаков вводятся с панели ввода греческих букв. Для задания приоритета в выполнении операций над множествами можно использовать круглые скобки. Множества могут иметь ряд эквивалентных форм, известных из теории множеств.

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 такой функции довольно проста: если условие выполняется, она возвращает одно значение, а если условие не выполняется — то другое.

Синтаксис задания условной функции IF в Derive следующий:

IF (Условие, u, v)

Если условие Условие дает TRUE, то эта функция возвращает значение выражения и, а если условие дает FALSE — то возвращается значение выражения V.

В общем случае можно использовать следующие конструкции условной функции:

IF (r,t,f) — если отношение r истинно, то выдает выражение t; если оно ложно, то выдает выражение f;

IF (r,t,f,u) — если отношение r истинно, то выдает выражение t; если оно ложно, то выдает выражение f; а если истинность неизвестна, то выдает и;

IF (r AND s,t,f) — если отношение r ложно, выдает выражение f; в противном случае выдает IF(s,t,f);

IF (r OR s,t,f) — если отношение r истинно, выдает выражение t; в противном случае выдает IF(s,t,f);

IF (NOT r,t,f) — если отношение r ложно, выдает выражение t; если оно истинно, выдает 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: $\dfrac{6842}{8131}$

5: F (-1)

6: 0

7: SIN (-1)

8: $-\dfrac{6842}{8131}$

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

Функции для организации итераций

Функции ITERATES и ITERATE

В практике программирования широко используются структуры, именуемые циклами. При этом различают циклы с заданным числом повторения операций внутри их и с числом операций, зависящим от выполнения некоторого условия. Оно обычно меняется 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 последовательностей данных. Остается необходимость в создании итерационных алгоритмов.

Такие алгоритмы в Derive реализует функция **ITERATES:**

ITERATES (u,x,x0,n) — выполнение п итераций при начальном значении x=xO 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 и(х) = х. Если указан параметр п (число итераций), то функция **ITERATES** проводит п итераций. Вначале она подставляет в (и(х) — х) начальное значение х = хО и выполняет первую итерацию, используя простой метод итераций. Затем полученное значение х → **x1** вновь подставляет-ся в (и(х) - **х)**, выполняется вторая итерация и т. д. Таким образом получается вектор значений х **(х0, х1,...)**.

Для уравнений и(х) - х = 0, при решении которых метод простых итераций сходится при достаточно большом п, можно получить ряд значений **х**, сходящихся к решению.

Если параметр п не указан, то итерации проводятся до тех пор, пока значения х 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 х.

Примеры реализации итерационных вычислений

Действие функций **ITERATES** и **ITERATE** поясняет следующий пример:

```
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 этом примере решается нелинейное уравнение

SIN(x) + 0.25 = x.

Нетрудно заметить, что функция **ITERATES** дает вывод всех значений х в ходе итераций, тогда как функция **ITERATE** возвращает только одно последнее перед остановкой итераций значение.

Реализация итерационного метода Ньютона

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: $$\text{NEWTON}(u, x, x0, n) := \text{ITERATE}\left(x - \frac{u}{\frac{d}{dx}u}, x, x0, n\right)$$

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 и'(x) обеспечивает высокую скорость сходимости метода Ньютона.

Другой пример показывает задание функции, решающей систему нелинейных уравнений:

FIXED_POINT(g,x,x0,n) := ITERATES(LIM(g,x,xk),xk,x0,n)

Здесь *g* — вектор правых частей системы нелинейных уравнений вида f(x)=x, **x** — вектор переменных, xO — вектор начальных значений переменных и п — число переменных.

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 может служить рекурсивная функция задания факториала:

FACT(n) := IF(n=0, 1, n FACT (n-1))

Она соответствует рекурсивному определению факториала

п! = 1 если n=0

n! = n (n-1)! если п>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 п 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 могут потребовать 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.

Практические приемы программирования

Пример вычисления мольной теплоемкости по Дебаю

Выше мы рассмотрели синтаксис Derive и полный набор встроенных операторов и функций системы. Однако само программирование рассматривалось на крайне примитивном уровне — примерах работы отдельных функций. В этом 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=@/Tотношение температуры Дебая @ к температуре T"

3: $$CV\ (z) := \frac{9\ r}{z^3} \int_0^z \frac{x^4\ EXP\ (x)}{(EXP\ (x) - 1)^2}\, dx$$

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 Т. Результат представляется как произведение числового коэффициента на универсальную газовую постоянную **r**.

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 не сразу понятная формулировка становится ясной из следующего простого примера:

```
1: "Функция пользователя с параметром — функцией"

         F (y, x, a) := LOG (lim    y)
2:                           x->a

              2
3:       F (x  , x, 2)

4:       1.38629

5: "Проверка"

                2
6:       LOG (2  )

7:       1.38629

              3
8:       F (x  , x, 2)

9:       2.07944

10: "Проверка"

                3
11:      LOG (2  )

12:      2.07944
```

В этом простом, но отнюдь не тривиальном примере задана некоторая функция **F(y,** x, a). Однако здесь y и x — необычные параметры. Они позволяют передать в правую часть выражения #2 некоторую произвольную функцию y(x), указанную обобщенным именем y, и имеющей аргумент x. А вот a — обычный параметр. Данная функция вычисляет натуральный логарифм для любого значения y(x) при x=a. Для этого несколько необычно использована функция вычисления предела — в данном случае 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 этой части могут стоять лишь определенные ранее (т. е. встроенные) функции. В этом примере параметры y и x передают в тело функции **F(y,x,a)** вначале функцию **x^2**, а затем функцию **x^3.**

Задание функции для вычисления интеграла по формуле Уэддля

Вполне полезным может служить другой пример — вычисление определенного интеграла вида

```
                      /a+h
INT_U(f(x),x,a,h) =  |       f(x)  dx
                     / a
```

по формуле Уэддля [1, **2]**, дающей высокую точность при разбивке интервала интегрирования от a до **b=a+h** на 6 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 Уэддля"

                       3h                                                ~
2:       INT_U (y,x,a,h) :=───((lim   y)  +  5       lim    y  +  (       lim    ~
                       10    x->a                x->a+h             x->a  +    ~

         ~            ~
         ~    y) + 6     lim    y + (    lim     y) + 5    lim     y +    ~
         ~-2  h        x->a+3  h       x->a+4  h        x->a+5  h    x->~

         ~
         ~lim     y)
         ~a+6  h

                 /                            1  \
3:       INT_U   |SQRT (2  x  +  1), x,  0,───|
                 \                            6 /

4:        1.39871

5:  "Проверка"
```

```
            1
           /
6:        /   SQRT (2 x + 1) dx
            0

7:        1.39871
```

Минимизация функций

Derive не содержит встроенных функций минимизации функции одной или многих переменных. Пожалуй, это главный 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 EXP (-x)

                          / d            \
3:        xmax := SOLVE  | —— F (x), x |
                          \ dx           /

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** показана простейшая проверка, показывающая, что значение x=1 в данном случае действительно есть значение, при котором функция F(x) имеет максимум, а не минимум, 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"

                                         / d              \
2:      X_EXTR (y, x, a, b) := SOLVE  |——y, x, a, b|
                                         \ dx             /

3:      X_EXTR (x EXP (-x), x, 0.5, 1.5)

4:       [x = 1]

                 / SIN (x)           \
5:       X_EXTR |————————, x, -1, 1|
                 \   x               /

6:       [x = 0]

7:      X_EXTR (SIN (x), x, -2, -1)

8:       [x = -1.57080]
```

В этом примере показано вычисление максимума для функций x*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 надежнее, быстрее и эффективнее.

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
```

$$3: \quad EI(m, x) := ce + LN(x) + \sum_{k=1}^{m} \frac{X^k}{k\ k!}$$

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 т=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 удобный для их поиска и идентификации). Что касается переходов по строкам (особенно безусловных), то они давно признаны 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.

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). Для этого известны три разные формулы для коэффициентов Берга, дающих относительную амплитуду, при п, равных **0,** 1 и п>=2. Ниже показано конструирование функции **BERG(n,t),** вычисляющей коэффициенты Берга при любых целых п и углах t (в радианах):

1: "Вычисление коэффициентов Берга"

$$2: \quad AO(t) := \frac{SIN(t) - t\ COS(t)}{pi\ (1 - COS(t))}$$

$$3: \quad A1(t) := \frac{t - SIN(t)\ COS(t)}{pi\ (1 - COS(t))}$$

$$4: \quad AN(n, t) := \frac{\frac{2}{Pi}(SIN(n\ t)\ COS(t) - n\ COS(n\ t)\ SIN(t))}{n\ (n^2 - 1)\ (1 - COS(t))}$$

```
5:      A01 (n, t) := IF (n > 0.5, A1 (t), AO (t))

6:      BERG (n, t) := IF (n > 1.5, AN (n, t), A01 (n, t))

                /   pi \
7:      BERG   |0, ---- |
                \   6  /

8:      0.110598

                /   pi \
9:      BERG   |1, -- - |
                \   6  /

10:     0.215223

                /   Pi \
11:     BERG   |2, -- - |
                \   2  /

12:     0.212206
```

В этом примере в строках **2—4** заданы три исходные функции АО, А1 и AN, вычисляющие коэффициенты Берга для **n=0,** 1 и любых п, равных или больше 2. В строке 5 определена функция **А01,** выбирающая с помощью функции IF функцию АО при п=0 или А1 при п=1. Далее в строке 6 определена искомая функция **BERG(n,t),** которая, также с помощью функции **IF,** выбирает функцию А01, если п меньше 2 и AN, если п больше или равно 2. Таким образом, основная функция BERG(n,t) вычисляет коэффициенты Берга при любых целых п и углах t — в строках **7—12** показаны примеры этого.

Функции выделения различных частей отношения

Некоторые системы компьютерной алгебры имеют богатый набор 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 г;

RHS **(r)** — возвращает правую часть отношения **r.**

Они могут быть полезны для программирования и решения некоторых задач символьной математики.

Расширенные средства программирования в Derive 5

Зарезервированные слова

Derive 5 имеет ряд зарезервированных слов:

Angle	Any	Approximate
ArrowKeyMode	Asterisk	Auto
Binary	Branch	CaseMode
Character	Collect	Complex
Compressed	Cosines	Decimal
Degree	**DisplayFormat**	Dot
Exact	exit	Expand
Exponential	Hexadecimal	Implicit
InputBase	Input**Mode**	Insensitive
Integer	**LineEdit**	Logarithm
Logical	Mixed	Nonscalar
Normal	Notation	NotationDigits
Octal	**OutputBase**	Precision
PrecisionDigits	Principal	Radian
RaDical	Rational	Real
Scientific	**ScientificThreshold**	Sensitive
Set	**SimplifyRelations**	Sines
Squarefree	Subexpression	TimesOperator
Trigonometry	Trig**power**	Trivial
VariableOrder	Vector	Word

Эти слова являются элементами алфавита языка программирования Derive 5 и их нельзя использовать в качестве имен (идентификаторов) переменных и функций пользователя.

Структурированный вывод листинга программ

Derive обеспечивает возможность задания процедур с помощью конструкции **PROG(u1, u2,** ...) **и** циклов **LOOP(u1,** u2, ...). При этом система дает автоматически структурированный вывод листинга процедур и функций. Например, если в строке ввода введена следующая функция — процедура:

```
REV(x,y):=PROG(y:=[],LOOP(IF(x=[],RETURN(y)),y:=ADJOIN(FIRST(x),y),
x::=REST(x)))
```

то в документе она предстанет в виде

```
REV(x, y) :=
  Prog
    y := []
    Loop
      If x = []
        RETURN y
    y := ADJOIN(FIRST(x), y)
    x := REST(x)
```

Такой вид соответствует канонам структурного программирования, но только внешне. Истинное структурное программирование основано на подготовке программ из полностью завершенных модулей. В Derive 5 такими модулями являются функции и реализуется функциональное программирование.

Если ввести определение переменной y в параметры функции, то ее можно представить в виде

```
REV(x, y:=[]) :=
  Loop
    If x = []
      RETURN(y)
    y := ADJOIN(FIRST(x), y)
    x := REST(x)
```

Множество примеров прекрасно составленных процедур и функций для Derive 5 можно найти в поставляемых вместе с системой файлах пакетов расширений.

Заключительные замечания по программированию

Ограничимся этими примерами функционального программирования на входном языке Derive, как достаточно типовыми. Следует отметить, что прилагаемая с Derive библиотека функций, описанных в дальнейшем (начиная с урока 11), является множеством прекрасно подобранных примеров на практическое программирование в среде Derive. Они наглядно показывают, что с помощью весьма ограниченных средств задания управляющих структур Derive способен решать довольно сложные математические задачи — от вычисления численными методами различных специальных функций до решения систем дифференциальных уравнений современными численными и аналитическими методами.

Завершая главу по 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 эффектом, 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 неожиданных эффектов непредусмотренного изменения значений переменных при исполнении той или иной функции.

Чему мы научились:

- Использовать алфавит системы и комментарии.
- Задавать основные типы данных.
- Использовать переменные и функции.
- Применять арифметические и логические операторы.
- Использовать функцию IF для создания условных выражений.
- Применять функции для организации итераций.
- Применять практические приемы программирования.
- Использовать функции выделения различных частей отношения.
- Применять расширенные средства программирования в Derive 5.

Урок 10. Встроенные функции

- Числовые функции
- Элементарные функции
- Функции комплексного аргумента
- Факториальные и комбинаторные функции
- Функции математического анализа
- Функции градиентного и векторного анализа
- Векторные и матричные функции
- Функции для множеств
- Статистические и финансовые функции
- Средства для проведения регрессии

Система Derive содержит довольно много встроенных **числовых,** арифметических и элементарных математических функций. Их несколько больше, чем это нужно для представления всего набора элементарных функций. Например, функция tan(x) 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.

Числовые функции

Числовые функции (x и y имеют действительные значения):

ABS (x) — возвращает абсолютное значение x;

SIGN (x) — возвращает 1 при x>0, 0 при x=0 и -1 при x<0;

MAX (x,y,...) — 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:

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

$$MIN\ (x,y)\ \to \frac{x+y}{2} \quad \frac{|x - y|}{2},$$

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

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

Примеры на использование этих функций даны 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 x STEP (x)

10: Y (1)

11: 5

12: Y (-1)

13: 0

14: F (x) := 5 x 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

Кусочно-непрерывные функции

Кусочно-непрерывные функции это функции, значения которых могут резко отличаться в точках разрывов, но в промежутках между ними они меняются 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 (т, п) — возвращает m по модулю п (неотрицательный остаток от m/n);

MOD **(m)** — возвращает дробную часть m;

MODS **(m,** п) — возвращает симметричное m по модулю п, которое оказывается в полуоткрытом интервале [-п/2,п/2), причем по умолчанию п=1;

GCD (m, п, ...) — возвращает наибольший общий делитель m, п, ...;

LCM **(m,** п, ...) — возвращает наименьшее общее кратное m, п, ...

Если т и п н е числа, то функции MOD ***(m,n)*** и MODS ***(m,n)*** упрощаются к следующим выражениям:

MOD (m,n) -> m - n FLOOR (m,n)

MODS (m,n) -> m — n FLOOR (m + n/2,n)

Элементарные функции

Степенные (экспоненциальные) функции

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

SQRT (z) — квадратный корень из z (на дисплее отображается как Vz);

#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

$$\#e^{z}\ \#e^{w}\ <->\ \#e^{z+w} \qquad (9.1)$$

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

$$(\#e)^{z\ k}\ <->\ \#e^{k\ z} \qquad (9.2)$$

справедливо, если **к** — целое или если к — 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 — (9.1) слева направо, а (9.2) 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 рекомендуется практически поупражняться в различных преобразованиях, используя разные их направления.

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 следующие преобразования:

$$LN(x)+LN(z)\ <->\ LN(x\ z) \qquad (9.3)$$

$$k\ LN(x)\ <->\ LN(x^{k}) \qquad (9.4)$$

Направления преобразования следующие:

Auto — (9.3) слева направо, а (9.4) 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.

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, SIN(z deg), где z — угол, заданный в градусах.

При преобразованиях тригонометрических функций действуют соотношения

```
TAN(z)=SIN(z)/COS(z)        COT(Z)=COS(z)/SIN(z)

SEC(z)=1/COS(z)             CSC(z)=1/SIN(z)
```

При упрощении выражений с тригонометрическими функциями используется команда **Manage Trigonometry.** Она дает 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):

```
SIN(z+w) -> SIN(z) COS(w) + COS(z) SIN(w)
```

Аналогично

```
SIN(m z) -> 2 COS((m-1) z) SIN(z) + SIN((m-2) z),
```

где m — положительное целое, большее 1.

Используйте **Toward Sines** для преобразований вида

```
      2              2
COS(z)  -> 1 - SIN(z)
```

и **Toward Cosines** для пребразований вида

$$SIN(z)^2 \to 1 - COS(z)^2$$

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

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 диапазон от -PI до +PI. Исключением является угол, возвращаемый функций **ATAN (y,x)** — он лежит в области значений -PI/2 до +PI/2. В этом же диапазоне углов лежат значения **ASIN (x),** если -1 <= x <= +1. При тех же значениях x значение **ACOS (x),** лежит в пределах от 0 до PI. При преобразованиях данного класса функций действуют следующие соотношения:

ACOT (z)=PI/2 - ATAN(z) ACOT (x,y)=ATAN(y,x)

ATAN (y,x)=ATAN(y/x) ACOS (z)=PI/2 - AS!N{z)

ASEC (z)=ACOS(1/z) ACSC (z)=ASIN <1/z)

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

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:

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

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

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:

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

Хорошо известны подобные соотношения и для других обратных гиперболических функций. Их можно найти в справке по системе Derive под Windows.

Примеры на операции с **элементарными функциями**

Ниже представлен ряд примеров на символьные и числовые вычисления с применением элементарных функций:

1: "Примеры операций с элементарными функциями:"

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

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

4: SIN (a + #i b)

5: $$\frac{\#e^{b}\ \mathrm{SIN}\ (a)}{2} + \frac{\#e^{-b}\ \mathrm{SIN}\ (a)}{2} + \#i\left(\frac{\#e^{b}\ \mathrm{COS}\ (a)}{2} - \frac{\#e^{-b}\ \mathrm{COS}\ (a)}{2}\right)$$

6: LOG (1000, 10)

7: 3

8: LN (a + #i b)

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

10: 10 (1 - EXP (-2))

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

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

Функции комплексного аргумента

Функции комплексного аргумента:
#i — мнимая единица **(Alt-I);**
ABS (z) — модуль z;

SIGN (z) — упрощается к точке на единичной окружности с углом тем же, 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.

Следующие примеры иллюстрируют применение этих функций:

```
1:"Функции комплексных чисел:"

2:      z := 3 + 2 #i

3:      SIN (z)

4:      0.530922 - 3.59056 #i

5:      RE (SIN (г))

6:      0.530922

7:      IM (SIN (z))

8:      -3.59056

9:      SIGN (z)

10:     0.832050 + 0.554700 #i

11:     |z|

12:     3.60555

          z
13:     ———
         |z|

14:     0.832050 + 0.554700 #i

15:     PHASE (z)

16:     0.588002

17:     CONJ (z).

18:     3 - 2 #i
```

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

Факториальные и комбинаторные функции

Факториальные и комбинаторные функции:
z! — факториал z;
GAMMA (z) или **Г(z)** —г гамма-функция;
PERM (n, m) — число размещений из п элементов по т;
COMB (n, т) — число сочетаний из п элементов по m.

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

z!=1 при z=0 или 1
z!=1*2*3*...*z при z>1.

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

z!=Г(z+1)

Поэтому Derive может вычислять z! по этому выражению и для любых z.
Функции **PERM** и **COMB** возвращают следующие значения:

$$PERM = \frac{п!}{\{n-m)!} \qquad COMB = \frac{п!}{<n-m)!*m!}$$

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 число специальных математических функций определено во внешних расширениях системы с помощью файлов ее библиотек (уроки 11 и 12). Нередко эти функции являются решениями дифференциальных уравнений специального вида, значениям интегралов или определениями специальных математических функций, дополняющих встроенные функции.

Функция создания таблицы истинности

Таблица истинности показывает логические значения Булевых выражений для всех возможных комбинаций значений переменных этого выражения. Функция **TRUTH_TABLE** в Derive 4.* под Windows создает таблицу истинности в виде матрицы для одной или более логических переменных. Если **p1,** p2, ... — логические переменные, то

TRUTH_TABLE(p1,p2 ... boolean1, boolean2,...)

создает таблицу истинности логических выражений в виде матрицы. Например,

TRUTH_TABLE(p, q, p AND q, p OR q, p XOR q, p IMP q)

упрощается к

P	q	p AND q	p OR q	p XOR q	p IMP q
true	true	true	true	false	true
true	false	false	true	true	false
false	true	false	true	true	true
false	false	false	false	false	true

Функции математического анализа

Обширный набор функций в системе Derive решает типовые задачи математического анализа [17, 18]. В основном это функции, реализующие типовые задачи высшей математики: определение пределов функций, вычисление производных и интегралов, разложение функций в ряд Тейлора и др. Надо полагать, что читатель, взявшийся за вычисление функций математического анализа, имеет о них достаточное представление. Иначе возможна ненормальная ситуация, как в известной поговорке «беда, коль пироги начнет печи сапожник, а сапоги точать пирожник».

Ниже представлены форматы использования этих встроенных функций. При этом введены следующие обозначения:

и — математическое выражение, описывающее функцию одной или многих переменных;

х — текущая переменная (например по которой идет дифференцирование или интегрирование);

п — порядок производной или первообразной;

a,b — значения х или пределы интегрирования.

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

Функции для вычисления пределов

Функции для вычисления пределов:

LIM (u,x,a) — предел функции и по переменной х в точке а;

LIM (u,x,a,1) — предел функции и по переменной х выше точки а;

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

Если задать

LIM (((x+h)^2-x^2)/((x+h)-x,h,0),

то на экране эта функция предстанет как

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

и упрощается к 2х (производная от **x^2).**

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

LIM (a x/(x+1), x, inf)

изображается как

```
         ax
lim ————————
x->oo   x + 1
```

и упрощается к а.

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

```
          x
lim ————————
x->0   SIN(x)
```

упрощается к 1.

Функции дифференцирования

Для дифференцирования используются функции, возвращающие:

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

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

Например, введя

```
DIF (SIN(a x^2),x)
```

получим на экране выражение

```
 d              2
——— SIN  (a  x   )
 dx
```

которое упрощается к

```
                2
2 a x COS (a x   )
```

Возможно последовательное применение функции **DIF** для вычисления смешанных производных.

Функция разложения в ряд Тейлора

Для вычисления разложения функции в ряд Тейлора используется функция

TAYLOR (u,x,a,n) — разложение функции и в n членов ряда Тейлора при x=a.

Она порождает степенной многочлен со степенями, убывающими от n до 0.

Функции интегрирования

Ряд функций осуществляют операции интегрирования:

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

DIF (и,х,-п) — 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 п со знаком «минус» возвращает первообразную, а не производную.

Функции суммирования и произведения членов рядов

Следующие функции используются для вычисления сумм и произведений:

SUM (u,n) — сумма ряда функции и по п;

SUM (u,n,k,m) — сумма ряда функции и для п от к до т;

PRODUCT (u,n) — произведение ряда функции и по п;

PRODUCT (u,n,k,m) — произведение ряда функции и для п от к до т.

Примеры на применение этих функций были подробно описаны в разд. 1.6. Нет смысла повторять их. Учтите, что возможно последовательное применение их, например, для вычисления кратных интегралов.

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 и и v — арифметические выражения, a и b — пределы изменения текущей переменной и x — корни уравнения (либо и=0, либо u=v).

Функции для решения уравнений:

SOLVE (u,x) — вычисление x, при котором и=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, . . .] , [x1,x2,...]) — вычисление корней x1, x2, ... для системы линейных уравнений.

Ниже представлен пример, показывающий возможности функции **SOLVE:**

1: "Решение уравнений — функция SOLVE"

2: $\text{SOLVE}(x^2 - 3, x)$

3: $\left[x = \frac{11274}{6509}\right]$

```
                2
4:      SOLVE (x  = 3, x)

        /     11274 \
5:      |x = ------- |
        \     6509  /

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:

```
3 * a + 2 * b + c = 4
    a +     b + c = 1
    a - 2 * b + c = 3
```

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

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 т. д.). Derive позволяет вычислять вектор-градиент для функции n-переменных, представленной выражением expn. При этом необходимо указать эти переменные как элементы некоторого вектора v. Ниже представлены функции и операторы для вычисления параметров дифференциального вектора.

Вычисление параметров дифференциального вектора:

GRAD (expn) — градиент expn в системе координат с переменными x, y и z;

GRAD (expn,v) — градиент expn в системе координат с переменными из вектора v;

GRAD (expп,A) — градиент expп в системе координат A;

DIV (v,A) — дивергенция вектора v;

LAPLACIAN (expn,A) — дивергенция градиента expп;

CURL **(v,A)** — ротор вектора v.

Ряд функций Derive позволяет вычислять потенциал (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: $\mathrm{GRAD}\,(x^3 + y^2 + z)$

3: $\left[3\,x^2,\ 2\,y,\ 1\right]$

4: $\mathrm{GRAD}\,(x^3 + y^2 + z,\ [x,\ y,\ z])$

5: $\left[3\,x^2,\ 2\,y,\ 1\right]$

6: $\mathrm{DIV}\left[1,\ 2\,y,\ 4\,z^2\right]$

7: $8\,z + 2$

8: $\mathrm{LAPLACIAN}\,(x^3 + y^2 + 2)$

9: $6\,x + 2$

10: $\mathrm{CURL}\left[y^3,\ 2\,x^2\,z,\ v\right]$

11: $\left[-2\,x^2,\ 0,\ 4\,x\,z - 3\,y^2\right]$

12: $\mathrm{POTENTIAL}\left[x,\ 2\,y^2,\ 3\,z^3\right]$

13: $$\frac{x^2}{2} + \frac{2\ y^3}{3} + \frac{3\ z^4}{4}$$

14: $$\text{VECTOR\_POTENTIAL}\left[x,\ 2\ y^2,\ 3\ z^3\right]$$

15: $$\left[2\ y^2\ z,\ -\ x\ z,\ 0\right]$$

Описанные в этой главе функции свидетельствуют об обширных возможностях системы Derive. Они могут быть дополнительно расширены при использовании библиотек функций. Их описанию посвящена следующая глава.

Векторные и матричные функции

Векторные функции и операторы

Ниже перечислены основные векторные и матричные функции Derive, причем и — арифметическое выражение, к — переменная индекса (цикла), v и w — векторы.

Векторные функции и операторы:

[x1,x2, ...,xп] — задание вектора x из п элементов

VECTOR (u,k,n) — задание вектора значений и при к меняющемся от 1 до п с шагом 1;

VECTOR **(u,k,m,n)** — задание вектора значений и при к, меняющемся от m до п с шагом 1;

VECTOR (u,k,m,n,s) — задание вектора значений и при к, меняющемся от m до п с шагом s;

ELEMENT (v,n) — выделение n-го элемента вектора v;

DIMENSION (v) — размерность п вектора v;

v . w — скалярное произведение векторов;

CROSS (v, w) — векторное произведение двух трехэлементных векторов.

Ниже представлены примеры на выполнение 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, к, 5)

12: [y, 2 y, 3 y, 4 У, 5 y]

13: VECTOR (к x, к, 3)

14: $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 4 & 6 & 8 & 10 \\ 3 & 6 & 9 & 12 & 15 \end{pmatrix}$

15: VECTOR (к y, к, 3, 6)

16: [3 y, 4 y, 5 y, 6 y]

17: VECTOR (к 2, к, 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 вектора. С позиций программирования эта функция вполне заменяет циклы как целочисленные, так и с нецелочисленной управляющей переменной к. 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: $$Z(x, y) := x^2 y^3$$

4: $$\mathrm{VECTOR}\left(\left(\frac{d}{dx}\right)^k Z(x, y),\ k,\ 3\right)$$

5: $$\left(2\,x\,y^3,\ 2\,y^3,\ 0\right)$$

6: "Создание матрицы из 6 производных функции"

7: $$H(x, y) := x^4 y^3$$

8: $$\mathrm{VECTOR}\left(\mathrm{VECTOR}\left(\left(\frac{d}{dy}\right)^m \left(\frac{d}{dx}\right)^k H(x,y), k, 0, 2\right), m, 1, 2\right)$$

9: $$\begin{pmatrix} 3\,x^4\,y^2 & 12\,x^3\,y^2 & 36\,x^2\,y^2 \\ 6\,x^4\,y & 24\,x^3\,y & 72\,x^2\,y \end{pmatrix}$$

10: "Вычисление определенного интеграла при разных b"

11: $$\mathrm{VECTOR}\left(\int_0^b \mathrm{SQRT}(2\,x + 1)\,dx,\ b,\ 0.5,\ 1.5,\ 0.5\right)$$

12: [0.609475, 1.39871, 2.33333]

13: "Вычисление сумм членов ряда с разным их числом"

$$14:\quad \text{VECTOR}\left(\sum_{n=1}^{k}\frac{1}{n},\ k,\ 100,\ 500,\ 100\right)$$

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. Они перечислены ниже

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:

[[a, b], [c, **d]]** — задание матрицы с размером 2x2;

IDENTITY_MATRIX (n) — задание единичной матрицы п*п;

ELEMENT (A, j, k) — выделение элемента Aj,k матрицы A;

A . B — скалярное произведение матриц;

A" — транспонирование матрицы A (замена строк столбцами);

DET (A) — вычисление детерминанта матрицы A;

TRACE (A) — вычисление следа матрицы A (суммы — диагональных элементов);

A ^ -1 — инвертирование матрицы A;

ROW_REDUCE (A) — ступенчатая форма матрицы A;

ROW_REDUCE (A, B) — ступенчатая форма матрицы A, расширенная матрицей B;

CHARPOLY (A, **mu)** — вычисление характеристического многочлена A с коэффициентами, заносимыми в вектор **mu;**

EIGENVALUES (A, **mu)** — 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 — это квадратная матрица, у которой диагональные элементов равны 1, а остальные элементы это 0;

ступенчатая (эрмитовая) 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 то все элементы выше первого ненулевого в строке — нули;

характеристический многочлен матрицы — это 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"

```
              -1
17:         a

           / -1.79166    0.916666    -0.125 \
           |                                |
18:        |  1.58333   -0.833333     0.25  |
           |                                |

           \ -0.125      0.25        -0.125 /

19: "Создание единичной матрицы 3*3"

20:        m := IDENTITY_MATRIX (3)

           / 1  0  0 \
           |         |
21:        | 0  1  0 |
           |         |

           \ 0  0  1 /

22: "Скалярное произведение матриц"

23:        a . m

           / 1  2  3 \
           |         |
24:        | 4  5  6 |
           |         |

           \ 7  8  1 /

25: "Характеристический полином матрицы a"

26:        CHARPOLY (a, p)
             3       2
27:        -p  + 7 p  + 66 p + 24

28: "Собственные значения матрицы a"

29:        EIGENVALUES {a, v)

30:        [v = -5.07440]

31:"Функция ROW_REDUCE"

32:        ROW_REDUCE (a)
```

```
        / 1  0  0 \
        |         |
33:     | 0  1  0 |
        |         |
        \ 0  0  1 /
34:
        ROW_REDUCE {a, m)
                                  0.916666   -0.125 \
        / 1  0  0   -1.79166                        |
35:     | 0  1  0    1.58333     -0.833333    0.25  |
        |                                           |
        \ 0  0  1   -0.125        0.25       -0.125 /
```

Для получения ступенчатой формы не сингулярной матрицы А используется функция **ROW_REDUCE (A).** Функция **ROW_REDUCE (A,B)** соединяет две матрицы и возвращает ступенчатую форму для расширенной матрицы.

Эти функции можно использовать для решения матричных уравнений вида А X = В.

Например, если А и В матрицы

```
|1 2|  и  |3 4|
|5 6|     |7 8|
```

то применение ROW_REDUCE (A, В) упрощается к матрице

```
|1 0 -1 -2|
|0 1  2  3|
```

т. е. решение есть

```
|-1 -2|
| 2  3|.
```

Функция **DIMENSION(A)** возвращает размер матрицы как число ее строк.

Матричные операции в символьной форме

При упрощении и нормализации выражений, содержащих векторные и матричные операторы Derive использует следующие соотношения векторной и матричной алгебры:

```
a'' -> a

          1
[1/a]` -> —
          a`
```

```
(a+b)` -> a` + b`

(a.b)` –> b` . a'

a.(b+c) -> a.b + a.c

(b+c).a -> b.a + c.a

(a b).c -> a.(b.c)
```

$$\frac{1}{a.b} \to \frac{1}{b} \cdot \frac{1}{a}$$

$$DET\ [1/a] \to \frac{1}{DET\ a}$$

DET (a.b) -> DET (a) DET (b)

В ряде случаев большой интерес представляет возможность выполнения матричных операций в символьном виде. Пример с такими операциями дан ниже:

```
1: "Символьные операции с матрицами"

2: "Задание матрицы в символьном виде"

            /a   b\
3:     m := |      |
            \c   d/

4: "Транспонирование матрицы"

5:     m`

            /a   c\
6:     т := |      |
            \b   d/
7: "Инвертирование матрицы"

         -1
8:     m
```

9: $$\begin{pmatrix} \dfrac{d}{a\,d-b\,c} & -\dfrac{b}{a\,d-b\,c} \\[2ex] -\dfrac{c}{a\,d-b\,c} & \dfrac{a}{a\,d-b\,c} \end{pmatrix}$$

10: "Задание вектора"

11: $$v := \begin{pmatrix} e \\ f \end{pmatrix}$$

12: "Решение системы линейных уравнений m*x=v"

13: "Используйте далее команду Simplify"

14: $$x := \frac{v}{m}$$

15: $$\begin{pmatrix} e \\ f \end{pmatrix} \begin{pmatrix} \dfrac{d}{a\,d-b\,c} & -\dfrac{b}{a\,d-b\,c} \\[2ex] -\dfrac{c}{a\,d-b\,c} & \dfrac{a}{a\,d-b\,c} \end{pmatrix}$$

16: "Вычисление детерминанта матрицы m"

17: DET (m)

18: $a\,d - b\,c$

19: "Вычисление следа матрицы m"

20: TRACE (m)

21: $a + d$

22: "Задание матрицы с элементами — выражениями'

```
         /  1            2 z    \
         |  ----        ----    |
         |   2           2      |
         |  z  - 3      z  - 3  |
23: ms := |                      |
         |    7            6    |
         |  ---------------     |
         \  z - 2        z - 5  /
```

24: "Вычисление детерминанта матрицы ms"

25: DET (ms)

26: "Используйте команду Simplify"

```
                  2
        2 (7 z  - 38 z + 6)
27: ---------------------------
         2
       (z  - 3) (z - 2) (z - 5)
```

Следует отметить, что элементами матриц и векторов могут быть арифметические выражения. В этом случае символьные операции выполняются над ними (см. пример в строках 22—27). Результаты операций могут быть 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 принтером, чем созерцать на экране дисплея.

Для 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 новой версии Derive 3.11 максимальный размер матриц существенно выше и он зависит от емкости ОЗУ — заметим, что для современных ПК с процессорами класса Pentium минимальный объем ОЗУ составляет 8—16 Мбайт.

Функции для множеств

Для работы с множествами Derive имеет две встроенные функции:

POWER_SET(s) — упрощается ко всем подмножествам s. Например, POWER_SET({1, 2}) упрощается к {{}, {1}, {1, 2}, {2}};

POWER_SET(s,n) — упрощается ко всем подмножествам s, имеющим точно n элементов. Например, POWER_SET({1, 2}, 2) упрощается к {{1, 2}}.

Статистические и финансовые функции

Derive имеет достаточный набор функций для проведения статистических расчетов и типовой статистической обработки данных. Прежде всего отметим четыре функции, широко применяемые в теоретической статистике.

Функция генерации случайных чисел

Некоторые виды анализа и моделирования различных явлений и систем требуют применения случайных чисел. В Derive есть функция генерации случайных чисел с равномерным распределением RANDOM (n), которая в зависимости от значения аргумента п выполняет следующие действия:

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

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

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

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

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 т и среднеквадратическим 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:

```
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 erf(x)+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 в виде их перечисления. Ниже представлены основные из этих функций.

Основные статистические функции

Статистические функции:

AVERAGE (z1, ..., zn) — арифметическое среднее;

RMS (z1, ..., zn) — среднеквадратическое значение;

VAR (z1, ..., zn) — дисперсия;

STDEV(z1, ..., zn) — среднеквадратическое отклонение;

FIT (m) — обработка матрицы m методом наименьших квадратов;

FIT (v,A) — многомерная линейная регрессия вектора **v**, задающего аргументы и по матрице данных A.

В представленных ниже примерах иллюстрируется действие большинства из 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: $\dfrac{\mathrm{SQRT}(2)\ \mathrm{SQRT}(a^2 + b^2)}{2}$

9: VAR (a, b)

10: $\dfrac{(a - b)^2}{4}$

11: RMS (1, 1.1, 1.2, 1.3, 1.4)

12: $\dfrac{\mathrm{SQRT}(146)}{10}$

13: VAR (1, 1.1, 1.2, 1.3, 1.4)

14: $\dfrac{1}{50}$

Обширными возможностями по статистической обработке данных обладает функция FIT. Она будет рассмотрена отдельно чуть позднее.

Финансово-экономические функции

Для проведения финансово-экономических расчетов в Derive включен ряд функций. Все они ориентированы на расчет вкладов и базируются на применении формулы

$$v(1+i)^n + p\,(1+i\,t)\frac{(1+i)^n - 1}{i} + f = 0$$

со следующими параметрами:

i — процентная ставка;

v **(pval)** — настоящее (на данный момент) значение вклада;

f **(fval)** — будущее значение вклада;

п (прег) — число регулярных вложений;

t (time) — время платежа (необязательный аргумент);

p **(pmt)** — пай (вложение или изъятие).

Аргумент t (time) означает время платежа. Он задается;

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 при вычислениях.

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, надо отметить, что в финансовой литературе такое правило 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 на вклады:

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: $$\frac{P}{i} - \frac{P\,(i+1)^n}{i}$$

6: $$\mathrm{FVAL}\left(\frac{5.25\%}{12},\ 12\ 10,\ -250\right)$$

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)

          /   P   \
       LN |-------|
11:       \ f i + p/
       ------------
         LN (i + 1)

             / 5.25%             \
12:   NPER   |------, 250, 39344 |
             \  12               /

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 виде. Рекомендуется внимательно протестировать необходимые расчеты, прежде чем полагаться на их полную достоверность. А для этого и полезен вывод формул финансовых расчетов и сравнение их с формулами, которые используются в отчетах или литературе пользователя.

Средства для проведения регрессии

Основная функция для проведения регрессии FIT

Особая роль принадлежит отмеченной выше функции **FIT(m).** Кратко ее назначение заключается в обработке матрицы m методом наименьших квадратов. Под этим емким определением лежит возможность решения задач на регрессию различных зависимостей — от простейшей линейной регрессии до линейной и нелинейной регрессии общего вида. Задачи этого типа широко распространены в физических и научно-технических приложениях статистических методов и поэтому требуют достаточно подробного обсуждения.

Линейная регрессия

Начнем с простейшей и одной из наиболее распространенных задач — линейной регрессии. Пусть надо облако точек (x,y) представить линейной зависимостью

y(x)=a*x+b.

При этом требуется найти параметры a и **b**, при которых среднеквадратическая погрешность приближения минимальна.

Для решения этой 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*x+b.** Остальные строки должны содержать пары значений x и y, т. е. исходные данные (например, экспериментальные). **Вычисление** функции **FIT(m)** с помощью команды **approX** даст искомую зависимость. Следующий пример поясняет это:

1: "Реализация метода наименьших квадратов"

2: "Линейная регрессия"

$$3:\quad \mathrm{FIT}\begin{pmatrix} x & a\,x+b \\ 2 & 5.5 \\ 4 & 6.3 \\ 6 & 7.2 \\ 10 & 8.6 \\ 8 & 8 \end{pmatrix}$$

4: "Используем команду approX"

5: 0.395 x + 4.75

Итак, в данном случае облако точек исходной зависимости приближено функцией **0,369x+4,75.** При этом обеспечивается наименьшая среднеквадратическая погрешность приближения этой функцией, которая является прямой.

Логарифмическая регрессия

Столь же просто можно провести логарифмическую регрессию:

```
1: "Реализация метода наименьших квадратов"

2: "Логарифмическая регрессия"

            /x    a + b LOG (x, 10) \
            |                       |
            |1         1            |
            |                       |
3:      FIT |2         1.45         |
            |                       |
            |3         1.7          |
            |                       |
            \4         1.9          /
4: 0.645501 LN (x) + 0.999640
```

Облако точек в данном случае приближено логарифмической функцией 0,645501 1п(x) + 0,99964.

Полиномиальная регрессия

Другой пример показывает использование функции **FIT(m)** для полиномиальной регрессии:

```
1: "Реализация метода наименьших квадратов"

2: "Полиномиальная регрессия"

            /       3     2          \
            |x    a x + b x + c x + d|
            |                        |
            | 1           1          |
            |                        |
3:      FIT | 2           4          |
            |                        |
            |             8.5        |
            | 3                      |
            \ 4           8          /
```

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.

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,** ... обеспечивающие приближение облака исходных точек комбинацией функций

y(x) = a*f1 (x) + b*f2<x) + c*f3(x) + ...

с наименьшей **среднеквадратической** 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"

$$3:\quad \text{FIT}\begin{pmatrix} x & a\,x^2 + b\ \text{EXP}\ (x) + \dfrac{c}{x} \\ 1 & 4.73 \\ 2 & 14.5 \\ -0.5 & -1.15 \\ -10 & 100 \\ -1 & 0.368 \end{pmatrix}$$

$$4:\quad 1.31098\ \#e^{x} + 1.00101\ x^2 + \frac{0.980392}{x}$$

Здесь надо обратить внимание на то, что выбранная функция регрессии должна быть линейна относительно искомых параметров, но может быть нелинейной относительно переменной x.

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 т. 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"

$$3:\quad \mathrm{FIT}\begin{pmatrix} x & y & a\ x + b\ \mathrm{EXP}\ (y) + c \\ 1 & 1 & 9.43 \\ 2 & 1 & 10.4 \\ 1 & 2 & 18.8 \\ 2 & 2 & 19.8 \end{pmatrix}$$

$$4:\quad 2.00930\ \#e^{y} + 0.985000\ x + 2.97562$$

Не следует думать, что Derive непременно повторяет заданное уравнение регрессии. В действительности система пытается провести возможные аналитические преобразования этого уравнения. Это видно 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.

Нелинейная регрессия с выводом данных в графической форме

При решении задач регрессии и аппроксимации весьма полезно сопровождать их выводом результатов в графическом виде. Наглядность задач намного возрастает, если наряду с построением графика функции регрессии строятся узловые точки и видна реализация алгоритма решения задачи.

Рис. 10.1 показывает обычное решение задачи на нелинейную регрессию в среде Derive под MS-DOS. После ввода первых пяти строк с применением команды Author выполняется команда approX и получается уравнение рег-

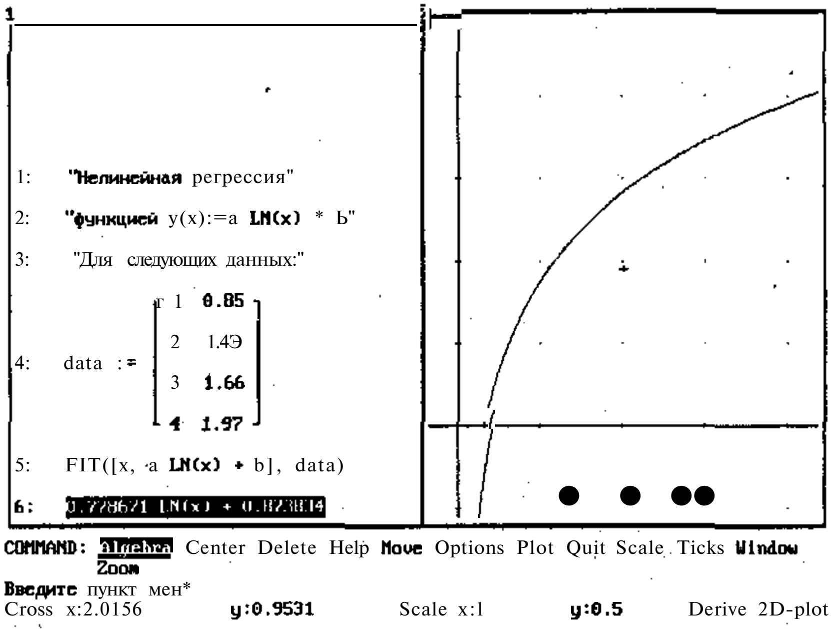


Рис. 10.1. Выполнение регрессии с построением графика ее функции

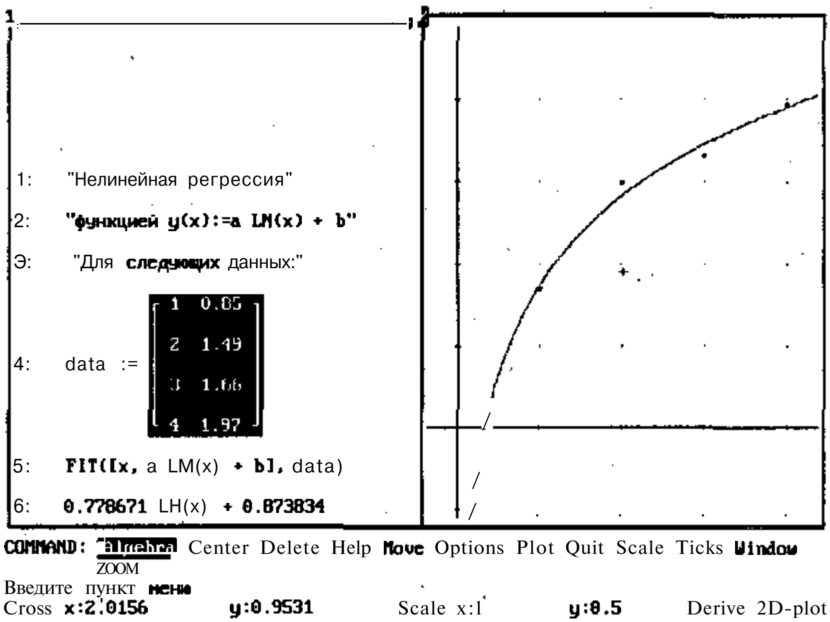


Рис. 10.2. Окно Derive после проведения регрессии и **построения** узловых точек (слева выделена матрица с координатами узловых точек) рессии — строка 6. Затем окно делится по вертикали и во втором окне (с использованием команды **Plot)** строится график функции регрессии.

Для построения узловых точек в окне графика нужно перейти вначале в окно с выражениями и выделить матрицу исходных данных. После этого, перейдя в окно графики, следует исполнить команду **Plot** — будут построены узловые точки с другим цветом, чем цвет линии регрессии. Это показано на рис. 10.2 (разумеется, без отличия цвета у линии регрессии и у узловых точек).

Рис. 10.2 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 принтером или направить в файл с расширением .TIF. При этом возможны эти операции как со всем экраном, так и с отдельными окнами.

Возможно также копирование экрана внешними программами (например, резидентной capture) с созданием файла с расширением **.PCX** и его последующим редактированием с помощью подходящего графического редактора, например Microsoft PaintBrush. Такое редактирование полезно, например, для обозначения кривых или узловых точек, нанесения цифр у делений масштабных осей и т. д.

Внимание

Если вы пользуетесь Derive под Windows, стоит повторить примеры по регрессии самостоятельно. Вы убедитесь в том, что это даже проще, чем при решении данной задачи в среде Derive под MS DOS.

Чему мы научились:

- Применять числовые функции.
- Использовать элементарные функции.
- Применять функции комплексного аргумента.
- Использовать факториальные и комбинаторные функции.
- Применять функций математического анализа.
- Применять функции градиентного и векторного анализов.
- Применять векторные и матричные функции.
- Использовать функции для множеств.
- Применять статистические и финансовые функции.
- Использовать средства для проведения регрессии.

Урок 11. Функции внешних расширений

- Дополнительные векторные и матричные функции
- Векторные и матричные операции — расширение vector.mth
- Специальные математические функции
- Графические функции — расширение graphics.mth
- Аппроксимация Паде — расширение approx.mth
- Прочие функции — расширение misc.mth
- Функции численного дифференцирования
- Функции численного интегрирования
- Вспомогательные функции
- Демонстрационные файлы (с расширением .dmo)

Дополнительные векторные и матричные функции

В этом уроке рассматриваются внешние расширения систем Derive, как ориентированных под MS-DOS, так и под Windows. Они поставляются вместе с системой в виде библиотечных файлов вида *.mth. В целом библиотеки Derive содержат определения сотен новых функций, большинство из которых едва ли понадобятся 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. разд. 3.10) тем, что-в их наименовании стоит знак (с) защиты авторских прав фирмы Soft Warehous Inc. Поэтому тексты этих файлов целиком в данной книге не приводятся. Пользователи могут ознакомиться с ними самостоятельно.

В новой версии Derive 5 под Windows загрузка библиотечных файлов сопровождается выводом сообщения об этом вида LOAD (name), где name — имя файла. В ранних версиях Derive такого сообщения не было. После загрузки библиотечных файлов находящиеся в них определения новых функций становятся доступными для применения.

К сожалению, в фирменную поставку системы не входят демонстрационные примеры на применение библиотечных файлов (нет их практически и в книге [8]). 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 до нескольких секунд (при опции **Load** загрузка занимает больше времени, так как текст файла выводится на экран дисплея).

Векторные и матричные операции — расширение vector.mth

Векторные операции

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 ...». Основные векторные функции этого расширения перечислены ниже (v и **w** — векторы, А и В — матрицы):

i_ — возвращает вектор [1, 0, 0];

j_ — возвращает вектор [0, 1, 0];

к_ — возвращает вектор [0, 0, 1];

OUTER (v,w) — возвращает Кронекерово (внешнее) произведение v и w;

KRONECKER (i,j) — возвращает дельта-функцию Кронекера, равную 1, если целое i равно целому j, и 0 в противном случае;

ADJOIN_ELEMENT (e,v) — присоединяет выражение е к началу вектора v;

APPEND_VECTORS (v,w) — объединяет вектор v с вектором **w;**

DELETE_ELEMENT (v,k) — уничтожает k-й элемента вектора v и уменьшает размер вектора на 1;

JACOBIAN (u,v) — возвращает Якобиан вектора **x=u(c),** криволинейные координаты с и Декартовы — х;

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.

Следующий пример иллюстрирует применение некоторых из этих функций:

```
1: "Действие функций файла vector.mth"

2:        L

3:        [0, 1, 0]

4:        v := [0, 1, 2, 3, 4, 5]

5:        OUTER (v, j_)
```

6: $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & O \\ 0 & 3 & 0 \\ 0 & 4 & 0 \\ 0 & 5 & 0 \end{pmatrix}$

7: KRONECKER (2, 2)

8: 1

9: KRONECKER (0, 1)

10: O

11: APPEND_VECTORS (v, v)

12: [O, 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]

Матричные функции

Кроме описанных выше векторных функций, файл **vector.mth** содержит определения ряда дополнительных матричных функций, где v и w — векторы, A и B — матрицы:

ADJOINT (A) — возвращает присоединенную или Эрмитово сопряженную матрицу **(A^-1*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 **approX);**

EXACT_EIGENVECTOR (A,mu) — возвращает собственный вектор A < 4x4 для собственного значения **mu;**

COVARIANT_METRIC_TENSOR (A) — возвращает ковариантный метрический тензор Якобиана матрицы A.

В файле **VECTOR.MTH** для использования вместе со встроенными функциями векторного анализа определены следующие полярно-цилиндрические и сферические координатные геометрические матрицы:

$$\text{cylindrical} := \begin{vmatrix} r & q & z \\ & & \\ 1 & r & 1 \end{vmatrix}$$

и

$$\text{spherical} := \begin{vmatrix} r & q & F \\ & & \\ 1 & r\cdot \mathrm{SIN}(F) & r \end{vmatrix}$$

Следующий пример показывает работу с некоторыми из этих матричных функций:

1: "Матричные функции файла **vector.mth**"

```
            / 1  2  3 \
            |         |
2:    a :=  | 4  5  6 [
            I   .     I
            \ 7  8  1 /

3:    FORCE0 (a, 2, 2, 1)

      / 1    2    3 \
      |             |
4:    | 1.5  0  -1.5 |
      |             |

5:    \ 7    8    1 /

      PIVOT (a, 2, 2)
                      |
6:    / 1    2    3  \|
      |               |

      |  4   5    6

      \ 0.6| 0  -8.6 /|
7:    b := | 5  6  7  |
           |/ 7  8  9\|
            \ 2  3  4/

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])

       / d     d     d   \
       | — 1   — 1   -  1|
       | d4    d5    d6  |
       |                 |
       | d     d     d   |
19:    | -  2  -  2  -  2|
       | d4    d5    d6  |
       |                 |
       | d     d     d   |
       |  - 3  -  3  -  3|
       \ d4    d5    d6  /

20:    COVARIANT_METRIC_TENSOR (a)

       / 66  78  34 \
       |            |
21:    | 78  93  44 |
       |            |
       \ 34  44  46 /
22:    GEOMETRY_MATRIX {a, b)

       / [1, 2, 3]  [4, 5, 6]  [7, 8, 1] \
23:    |                                 |
       \ 2.64575    2.44948       2      /

24: cylindrical

       / r  theta  z \
25:    |             |
       \ 1    r    1 /
26: spherical
       / r      theta        phi \
27:    |                         |
       \ 1   r SIN (phi)     r   /
```

Действие большинства из этих функций относится к специальным разделам векторного и матричного анализов, включая тензорный анализ и теорию упругости. Их применение заметно расширяет возможности системы Derive в проведении специальных операций с векторами и матрицами. К сожалению, характер и объем данной книги не позволяет остановиться на более детальном описании применения данных функций. Остается уповать на то, что те, кому это нужно, разберутся с более тонкими деталями применения описанных функций.

Специальные математические функции

Средства Derive позволяют вычислять практически все специальные математические функции по алгоритмам, описанным в литературе [16]. Для этого необходима загрузка соответствующих файлов. Их имена и вводимые с их помощью функции указаны ниже. Для ряда функций вычисление реализована 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 функции стоит слово series (ряд). Для каждого класса функций даны контрольные примеры.

Интегральные показательные функции (exp_int.mth)

Интегральные показательные функции в файле **exp_int.mth** содержат следующие определения:

```
euler_gamma:=0.5772156649015328606065l2
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_)/(-z)^l_,l_,0,m)
E1(z,m):=-euler_gamma-LN(z)-SUM((-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:

euler_gamma — константа Эйлера (25 точных цифр);

EI(x,m) — интегральная показательная функция **INT(#e^-t/t,t,-x,inf)** для; x>0, вычисленная по ее разложению в ряд;

LI(x,m) — логарифмическая интегральная функцию **INT(1/LN t,t,0,x)** для **x>1,** вычисленная по ее разложению в ряд;

EN (n,z) — интегральная показательная функция n-го порядка **INT(exp(-zt)/t^n,t,1,inf)** при **RE(z)>0, n>=0;**

EN_ASYMP (n,z,m) — асимптотическое приближение **EN(n,z)** для больших |z|, вычисляемое по разложению в **ряд,** ограниченный т + 1 членами;

E1(z,m) — показательная функция **E1(z)=EN(1,z),** вычисляемая по ее разложению в ряд;

SI(z) — интегральный синус **INT(SIN(t)/t,t,0,z);**

CI(z) — интегральный косинус **INT(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

Дополнительные вероятностные функции (probabil.mth)

Дополнительные вероятностные функции, широко применяемые в специальных статистических расчетах, заданы в файле probabil.mth:

POCHHAMMER (a,x) — функция (a)x = GAMMA(a+x)/GAMMA(a);

PSI (z) — пси-функция DIF(GAMMA(z),z)/GAMMA(z) для модуля аргумента **z**, не превосходящего pi/2;

POLYGAMMA (n,z,m) — полигамма — функция DIF(PSI(z),z,n), где z/=0,-1,-2,..., вычисляемая по разложению в ряд;

INCOMPLETE_GAMMA (z,w) — функция P(z,w)=INT(#e^-t*t^(z-1),t,0,w)/GAMMA(z), RE(z)>0; .

INCOMPLETE_GAMMA_SERIES — функция P(z,w), вычисляемая по (z,w,m) разложению в ряд;

BETA (z,w) — бета-функция B(z,w)=GAMMA(z)*GAMMA(w)/GAMMA(z+w),

INCOMPLETE_BETA (x,z,w) — функция Bx(z,w) = INT(t^(z-1)*(1-t)^(w-1),t,0,x);

POISSON_DENSITY (k,t) — плотность вероятности распределения Пуассона #e^-t*t^k/k!;

POISSON_DISTRIBUTION(k,t) — распределение Пуассона, т. е. функция SUM (#e^-t*t^j/j!,j,0,k);

BINOMIAL_DENSITY (k,n,p) — функция COMB(n,k)*p^k*(1-p)^(n-k);

BINOMIAL_DISTRIBUTION (k,n,p) — биномиальное распределение, т. е. функция SUM (COMB(n,j)*p^j*(1-p)^(n-j),j,0,MIN(k,n));

HYPERGEOMETRIC_DENSITY (k,n,p) — гипергеометрическое распределение COMB(m,k)*COMB(j-m,n-k)/COMB(j,n);

HYPERGEOMETRIC_DISTRIBUTION (k,n,p) — функция кумулятивного гипергеометрического распределения;

STUDENT (t,v) — кумулятивная плотность распределения Стьюдента A(t|v);

F_DISTRIBUTION (f,v1,v2) — **F-функция** кумулятивного распределения P(f|v1,v2);

CHI_SQ (u,v) — функция Chi-квадрат распределения P(u|v), u = Chi^2.

В примере ниже 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)

9: $\dfrac{\#e^{k\ LN\ (t)\ -\ t}}{k!}$

10: POISSON_DENSITY (2, 0.5)

11: 0.0758163

12: HYPERGEOMETRIC_DENSITY (k, n, m, j)

13: $$\frac{n!\ m!\ (j - n)!\ (j - m)!}{(j + k - m - n)!\ j!\ k!\ (m - k)!\ (n - k)!}$$

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

Интегралы Френеля (fresnel.mth)

Интегралы Френеля **в** файле fresnel.mth 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)
```

Назначение этих функций следующее:

FRESNEL_SIN (z) — синусный интеграл Френеля **S(z)=INT(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)=INT(COS(pi*t^2/2),t,0,z);

FRESNEL_COS_SERIES (z,m) — косинусный интеграл Френеля, вычисляемый по разложению в ряд.

Ниже даны примеры на вычисление 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

Функции Бесселя и Эйри (bessel.mth)

Функции Бесселя и Эйри заданы в файле bessel.mth:

BESSEL_J (n,z) — функция Бесселя первого рода Jn(z) при рациональном n;

BESSEL_J_ASYMP (n,z) — асимптотическое приближение Jn(z) для больших |z|, вычисляемое по разложению в ряд;

BESSEL_Y **(n,z)** — функция Бесселя второго рода Yn(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 Yn(z) для больших [z|;

SPHERICAL_BESSEL_J(n,z) — сферические функции Бесселя первого рода в закрытой форме jn(z);

SPHERICAL_BESSEL_Y(n,z) — сферические функции Бесселя второго рода в закрытой форме yn(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 представлено 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_SER!ES (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

Гипергеометрические функции **(afqk** hipergeo.mth)

Гипергеометрические функции заданы в файле hipergeo.mth:

KUMMER (a,b,z) — конфлюентная 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.

Следующие примеры показывают 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

Эллиптические интегралы **(elliptic.mth)**

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 используются в следующем виде:

ELLIPTIC_F (phi,m) — эллиптический интеграл первого рода F(phi\m);

ELLIPTIC_E (phi,m) — эллиптический интеграл второго рода E(phi\m);

ELLIPTIC_PI **(phi,m,n)** — эллиптический интеграл третьего рода П(n;phi\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

Ортогональные полиномы (orth_pol.mth)

Ортогональные полиномы заданы функциями файла **orth_pol.mth:**

CHEBYCHEV_T (n,x) — полином Чебышева первого рода Tn(x);

CHEBYCHEV_U (n,x) — полином Чебышева второго рода Un(x);

LEGENDRE_P (n,x) — полином **Лежандра** Pn(x);

ASSOCIATED_LEGENDRE_P (n,m,x) — полином Лежандра **Pn^m(x);**

HERMITE_H (n,x) — полином Эрмита Hn(x);

HERMITE_HE (n,x) — ассоциированный полином Эрмита HEn(x);

WEBER_D (n,x) — цилиндрическая параболическая функция Вебера Dn(x);

GENERALIZED_LAGUERRE(n,a,x) — полином Лагерра в общей форме **Ln(n,a,x),** при **a=0** — ординарная форма;

JACOBI_P (n,a,b,x) — полином Якоби **Pn^(a,b)(x);**

GEGENBAUER_C (n,a,x) — ультрасферический полином Гегенбауэра **Cn^a(x).**

Задачи на вычисление ортогональных многочленов даны в следующем примере:

```
1: "Вычисление ортогональных полиномов"

2;        CHEBYCHEV_T (3, x)

                      2
3:        x (4 x  - 3)

4:        CHEBYCHEV_U (3, x)

                        2
5:        4 x (2 x  - 1)

6:        LEGENDRE_P (3, x)

                          2
7:        0.5 x (5 x  - 3)

8:        HERMITE_H (3, x)

                        2
9:        4 x (2 x  - 3)

10:       WEBER_D (3, x)

                2           /                     2                \
               x  / 4       |  -(SQRT (2) x/2)      SQRT (2) x    |
11:       - #e        DIF   |#e                  ,--------------,3|
                            \                          2          /
12:       GENERALIZED_LAGUERRE (3, 1, x)
                               3           2
13:       - 0.166666 (x  - 12 x + 36 x  - 24)
```

Дзета-функция, полилогарифм и дилогарифм (файл zeta.mth):

Дзета-функция, полилогарифм и дилогарифм определены в файле zeta.mth:

ZETA (s,m) — дзета-функция Римана **SUM(k^-s,k,1,inf),** вычисляемая по разложению в ряд;

HURWITZ_ZETA (s,a,m) — **Гурвица** функция **SUM((k+a)^-s,k,0,inf),** вычисляемая по разложению в ряд; -

POLYLOG (n,z,m) — полилогарифм Lin(z), вычисляемый по разложению в ряд;

LERCH_PHI (z,s,a,m) — пси-функция Phi(z,s,a), вычисляемая по разложению в ряд;

DILOG (x) — дилогарифм **INT(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)
```

$$11:\quad -\int_{1}^{x} \frac{LN(t)}{t-1}\,dt$$

```
12:      DILOG {0.5)

13:      0.582240
```

Из приведенного описания вытекает, что библиотеки Derive содержат богатый набор специальных математических функций, который вполне удовлетворит 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 **[16].** 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 вьюверов.

Внимание

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 функции 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. В этом отношении более мощные системы Maple V или Mathematica 2/3/4 куда более предпочтительны.

Графические функции — расширение graphics.mth

Файл содержит большое число 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 этих функций используется окно двухмерной графики.

Функции файла graphics.mth следующие:

ISOMETRIC (v) — матрица двухмерной изометрической проекции трехмерного координатного вестора v = [x(t),y(t),z(t)], представляющего трехмерную фигуру (используется с командой **Plot);**

ROTATE_X (phi) — матрица А такая, что A.[x,y,z] поворачивает фигуру на угол phi относительно оси x;

ROTATE_Y (phi) — матрица А такая, что A.[x,y,z] поворачивает фигуру на угол phi относительно оси y;

ROTATE_Z (phi) — матрица А такая, что A.[x,y,z] поворачивает фигуру на угол phi относительно оси z;

ISOMETRICS (v,s,s0,sm,m,t,t0,tn,n) — матрица поверхности [x(s,t), y(s,t),z(s,t)], s=s0...sm, t=t0...tn (используется команда **approX** 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=x0+#i*y0...** xm + #i, отображаемая в графическом окне в виде набора лучей на прямоугольной сетке, с длиной и направлением, соответствующим значению w;

HORIZONTALS **(w,z,z00,zmn, m,n)** — матрица, представляющая в двухмерном окне набор точек, где каждый набор значений w соединен отрезками прямых;

ARCS **(w,z,r0,rm,m,th0,thm,n)** — матрица, представляющая в двухмерном окне набор точек, где каждый ряд значений соединен сегментами.

Кроме того, в этом файле определена переменная axes, значением которой является матрица 3*2, параметрически задающая изометрическую проекцию осей X, Y и Z в трехмерной Декартовой системе координат. Эта переменная используется для графического построения осей. Примеры на вычисление некоторых графических функций представлены ниже:

```
1: "Некоторые графические функции"

2:      axes

        / -t_  - 0.5 t_\
        |              |
3:      |  t_  - 0.5 t_|
        |              |
        \  0       t_  /

4:      ROTATE_Z (a)

        / COS (a)  - SIN (a)   0 \
        |                        |
5:      | SIN (a)    COS (a)   0 |
        |                        |
        \   0          0       1 /

6:      CYLINDER (1, t, z)

7:      [COS (t), SIN (t), z]

8:      CONE (a, t, 2)

9:      [z COS (t) SIN (a), z SIN (t) SIN (a), z]
```

10: RE_IM (a + #i b)

11: [a, b]

Бо́льши́нство функций файла **graphics.mth** служит для вычисления координат точек различных графических построений, 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 **approX,** получите их значения. На рис. 11.1 они представлены в левом окне, правое окно создано для вывода графиков. Позаботьтесь о том, чтобы обычные оси были не видны — 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,

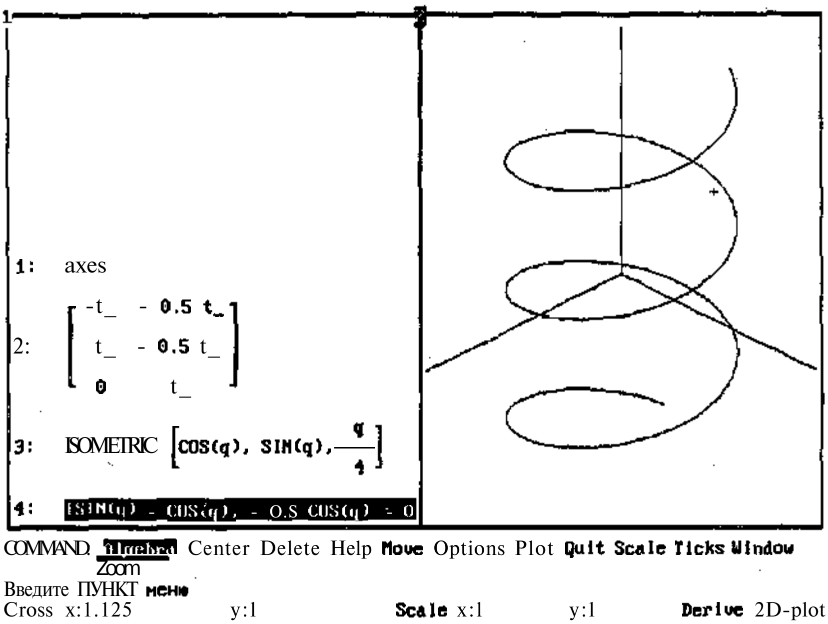


Рис. 11.1. Построение объемной спирали получим три оси (см. рис. **11.1).** Затем, выделив выражение в строке 4, построим спираль.

Поупражняйтесь с этим примером, пока не получите полного соответствия с рисунком, показанным на рис. 11.1.

Аппроксимация Паде: расширение — файл approx.mth

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 approx.mth. Он содержит (наряду с вспомогательными функциями) функцию **PADE:**

```
CF(a,m,n):=VECTOR(VECTOR(ELEMENT(a,m-n+i+j),j,1,n+1),i,1,n)
DV(a,m,n):=ELEMENT(ROW_REDUCE(CF(a,m,n))`,n+1)
NV(a,u,m,n):=VECTOR(ELEMENT(a,k+1)-SUM(ELEMENT(u,n+1-i)*ELEMENT~
(a,k+1-i),i,1,k),k,1,MIN(m,n))
P3(a,v,u,x,c,m,n):=(ELEMENT(a,1)+SUM(ELEMENT(v,k)*(x-c)^k,k,1,m))/~
(1-SUM(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 аппроксимация Паде 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 для функций **(x^2+1)** и sin(x):

1: "Аппроксимация Паде"

2: $\quad$ PADE $(x^2 + 1, x, 1, 3, 3)$

3: $\quad x^2 + 1$

4: $\quad$ PADE (SIN (x), x, 1, 2, 2)

$$5:\quad \frac{0.00306523\,(3.94074\cdot 10^{10}\, x^{2} - 1.27412\cdot 10^{11}\, x + 9.35916\cdot 10^{8})}{3.21290\cdot 10^{7}\, x^{2} - 9.08822\cdot 10^{7}\, x + 3.75921\cdot 10^{8}}$$

В строке 2 этого примера задана аппроксимация Паде 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. 11.2 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 аппроксимации Паде.

Необходимые для аппроксимации действия видны в левом окне. Там же находятся результаты аппроксимации. Выделив функцию TAN(x) и результаты ее аппроксимации, можно построить графики функций — они представлены на рис. **11.2** в правом окне.

При интерпретации нескольких графиков имеются определенные проблемы по их выделению. Derive строит графики с разными цветами линий, по-

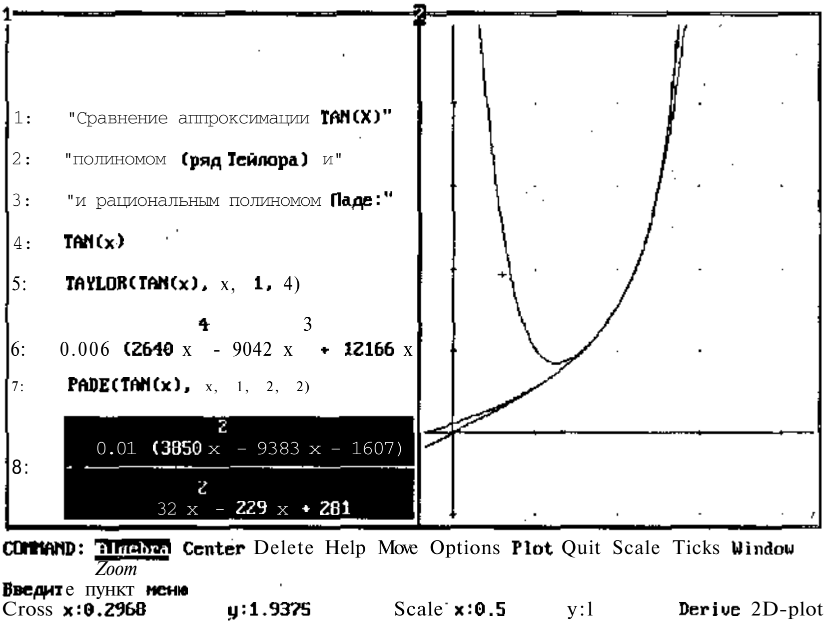


Рис. 11.2. Пример аппроксимации функции TAN(x) двумя методами

этому их можно идентифицировать по цвету. Однако при печати графиков обычными принтерами (не цветными) этот способ не годится.

Один из способов идентификации заключается в словесном описании. Например, на рис. 11.2 график полиномиальной аппроксимации рядом Тейлора помечен крестиком-меткой. График ниже — аппроксимация Паде, еще ниже — просто график функции TAN(x). Нетрудно заметить, что аппроксимация рядом Тейлора дает большие ошибки при приближении графика TAN(x) не только к точкам разрыва, но и к нулю. Аппроксимация Паде в этом отношении намного лучше. Если вы пользуетесь Derive под MS-DOS, то с помощью графического редактора Paintbrush можете нанести на графики наглядные указания и комментарии. В версии Derive 4.* под Windows предусмотрено нанесение на график в любом месте текстовых комментариев. А версия Derive 5, как отмечалось, позволяет плавно перемещать уже созданные комментарии мышью, точно позиционируя их на графике.

Прочие функции — расширение **misc.mth**

В файле misc.mth система Derive содержит определения ряда различных функций, которые трудно отнести к описанным ранее группам:

MOD (a,b) — деление a на b по модулю (дает не отрицательный остаток от a/b);

FLOOR (a,b) — наибольшее целое, меньшее или равное a/b;

RATIO_TEST (t,n) — если > 1, то SUM(t,n,a,inf) дает сумму конвергенций, а при < 1 — сумму дивергенций;

LIM2 (u,x,y,x0,y0) — предел u(x,y) при [x,y], стремящемся к [x0,y0] по всей линии наклона;

LEFT_RIEMANN (u,x,a,b,n) — левая сумма Римана для 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 INT (u*dv,x);

INT_SUBST (y,x,g) — действие simplify INT(y,x) с использованием подстановки x->g(x);

PROVE_SUM (t,k,a,n,s) — дает [0,0], если SUM(t,k,a,n)=s.

Ниже представлены примеры на вычисления с этими функциями:

```
1: "Вычисление функций файла misc.mth"

2:        MOD (10, 3)

3:        1

4:        FLOOR (a, b)
```

5: $\dfrac{\mathrm{ATAN}\left(\mathrm{COT}\left(\dfrac{\mathrm{pi}\ a}{b}\right)\right)}{\mathrm{pi}} + \dfrac{2\ a - b}{2\ b}$

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

Функции численного дифференцирования — расширение numeric.mth

В файле numeric.mth задан 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 представлены функции файла numeric.mth с указанием возвращаемого ими значения.

Функции численного дифференцирования

Для численного дифференцирование в файле numeric.mth имеются следующие функции:

DIF_NUM (y,x,x0,h) — дает первую частную производную y по x в точке xO 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 данными 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 $\left(\begin{pmatrix}0\\1.1\\2.1\\2.8\\4.1\\5.2\end{pmatrix}, 1\right)$

```
     /    0    \
     |          |
     |          |
     ! 1.06666  |
     |          |
9:   |    2     |
     |          |
     |    3     |
     |          |
     | 4.03333/
     |
     \  5.2
```

```
          /                 / 1  1 \\
          |                 |       ||
10:       | DIF_DATA        | 2  2  ||
 INT_DATA |                 |       ||
          \                 \ 3  3 //
```

```
     / 1  0 \
     |       |
11:  | 2  1  |
     |       |
     \ 3  2 /
```

В последних версиях Derive в файл number.mth добавлена функция **MERSENNE(n)**, возвращающая n-е число Мерсенна.

Применение операций дифференцирования файл **(dif_apps.mth)**

Применение операций дифференцирования реализуется следующими функциями файла:

CURVATURE (y,x) — кривизна зависимости y(x);

CENTER_OF_CURVATURE (y,x) — вектор центра кривизны линии y(x);

TANGENT (y,x,x0) — касательная к линии y(x) в точке x=x0;

PERPENDICULAR (y,x,x0) — перпендикуляр к линии y(x) в точке x=xO;

OSCULATING_CIRCLE (y,x,theta) — соприкасающаяся с линией y(x) окружность, заданная параметрически с параметром theta;

PARA_DIF (v,t,n) — n-я производная v по t, где v=[x(t),y(t)];

PARA_CURVATURE (v,t) — кривизна параметрической кривой v;

PARA_CENTER_OF_CURVATURE (v,t) — вектор центра кривизны v;

PARA_TANGENT (v,t,t0,x) — касательная к v при t=t0, выраженная через x;

PARA_PERPENDICULAR (v,t,t0,x) — перпендикуляр к v при t=t0;

PARA_OSCULATING_CIRCLE (v,t, t0,phi) — соприкасающаяся при t=t0 с v окружность, заданная параметрически с параметром phi;

POLAR_DIF (r,theta,n) — n-я производная r(theta), в функции от theta;

POLAR_CURVATURE (r,theta) — кривизна функции r(theta) в полярной системе координат;

POLAR_CENTER_OF_CURVATURE(r,theta) — центр кривизны функции в полярной системе координат;

POLAR_TANGENT (r,theta,theta0,x) — касательная к r(theta) при theta=theta0 в функции от x;

POLAR_PERPENDICULAR (r,theta, theta0,x) — перпендикуляр к радиусу окружности при theta=theta0;

IMP_DIF (u,x,y,n) — неявная производная DIF(y,x,n) для u(x,y)=0 при n целом положительном (по умолчанию равным 1);

IMP_CURVATURE (u,x,y) — кривизна для неявной кривой u=0;

IMP_CENTER_OF_CURVATURE(u,x,y) — центр для кривизны u=0;

IMP_TANGENT (u,x,y,x0,y0) — касательная линия к u=0 при x=x0 и y=y0;

IMP_PERPENDICULAR (u,x,y,x0,y0) — перпендикуляр u=0 при заданных x0 и y0;

IMP_OSCULATING_CIRCLE (u,x,y,x0,y0,phi) — соприкасающаяся в точке (x0,y0) окружность, заданная параметрически с параметром phi;

TANGENT_PLANE (u,v,v0) — плоскость касательная к u(x,y,z)=0 при [x,y,z]=v=v0;

NORMAL_LINE (u,v,v0,t) — декартовый координатный вектор, линейный по переменной t (нормаль к u=0 при v=v0).

Все эти функции возвращают аналитические выражения для того или иного геометрического понятия. Разумеется, используя подстановки, их можно в дальнейшем использовать и для численных расчетов. Применение функций дифференцирования поясняют следующие примеры:

1: "Применение некоторых функций дифференцирования"

2: CURVATURE (x^2, x)

3: $$\frac{2}{(4x^2+1)^{3/2}}$$

4: CENTER_OF_CURVATURE (x^2, x)

5: $$\left(-4x^3,\ 0.5\,(6x^2+1)\right)$$

6: TANGENT $(x^2, x, 1)$

7: $2x - 1$

8: $\mathrm{PERPENDICULAR}(x^2,\ x,\ 1)$

9: $-0.5\ (x - 3)$

10: $\mathrm{OSCULATING\_CIRCLE}(x^2,\ x,\ 0)$

11: $\left[0.5\,(4\,x^2 + 1)^{1.5} - 4\,x^3,\ 0.5\,(6\,x^2 + 1)\right]$

12: $\mathrm{POLAR\_DIF}(r,\ theta,\ 2)$

13: $-\dfrac{\mathrm{COS}(theta)^2}{r\ \mathrm{SIN}(theta)^3} - \dfrac{1}{r\ \mathrm{SIN}(theta)}$

14: $\mathrm{NORMAL\_LINE}(u,\ v,\ v0,\ t)$

15: $t\ \mathrm{GRAD}(u,\ 0)$

Большинство функций файла dif_apps служат для решения специальных геометрических задач. Ввиду очевидности их тестирование задач, не попавших в приведенные примеры, может быть выполнено заинтересованным пользователем самостоятельно.

Дополнительные функции интегрирования (файл int_apps.mth)

Файл int_apps.mth содержит большое число функций, базирующихся на численном интегрировании или представляющих интегральные преобразования специального вида. Имеется также много функций 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 данном файле представлены функции для расчета по интегральным выражениям площади (в названии слова 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 **x1, y1, z1** и x2, y2, z2 — указываются области ограничения фигур.

Ниже перечислены функции файла **int_apps.mth** с указанием возвращаемого ими значения.

FOURIER (y,t,t1,t2,n) — серия из n гармоник тригонометрического ряда Фурье, приближающего функцию y(t) в интервале t от t=t1 до t2;

LAPLACE (y,t,s) — интегральное преобразование Лапласа для функции y(t), дающее функцию от независимой переменной s;

ARC_LENGTH (y,x,x1,x2) — длина дуги функции y(x) для x в интервале от **x1** до x2;

ARC_LENGTH (y,x,x1,x2,mu) — интеграл mu(x) по всей линии дуги y(x) для x от x=xl до x2;

POLAR_ARC_LENGTH (r,th, th1,th2) — длина полярной дуги r(th) при изменении th от **th1** до th2;

POLAR_ARC_LENGTH (r,th, th1 ,th2,mu) — интеграл mu(th) **по** всей линии дуги **r;**

PARA_ARC_LENGTH (v,t,t1,t2) — длина дуги для вектора v(t) при t от t=tl до t2;

PARA_ARC_LENGTH (v,t,t1, t2,mu) — интеграл mu(t) **по** всей линии v при t от t=tl до t2;

AREA (x,x1,x2,y,y1,y2,mu) — интеграл **mu(x,y)** по области x для x от xl до x2 и для y от yl(x) до y2(x);

AREA (x,x1 ,x2,y,y1 ,y2) — площадь области при изменении x от x до x2 и y от yl(x) до y2(x);

AREA_CENTROID (x,x1 ,x2, y,y1 ,y2) — геометрический центр масс области при изменении x от xl до x2 и y от yl(x) до y2(x);

AREA_CENTROID (x,x1 ,x2, y,y1 ,y2,mu) — центр масс ограниченной области с распределением плотности mu(x,y);

AREAJNERTIA (x,x1,x2,y,y1,y2) — геометрический тензор инерции для ограниченной области;

AREAJNERTIA (x,x1 ,x2, y,y1 ,y2,mu) — тензор инерции для ограниченной области с распределения плотности **mu(x,y);**

POLAR_AREA (r,r1 ,r2, th,th1 ,th2) — площадь области в полярной системе координат при при th, меняющемся от thl до th2, и r от rl(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 xl до x2 и y от yl(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 xl до **x2,** y от yl(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 xl до x2, y от yl(x) до y2(x) и z от zl(x,y) и z2(x,y);

VOLUMECENTROID (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);**

VOLUMEJNERTIA (x,x1,x2,y,y1,y2,z,z1,z2) — объемный инерционный тензор для ограниченной области;

VOLUMEJNERTIA (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 thl(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,ph)** по ограниченной области в сферической системе координат, где th и ph — 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 thl(z) до th2(z), r от rl(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 **x1** до x2;

VOLUMEY_OF_REVOLUTION (y,x,x1,x2) — объем фигуры, полученной вращением линии y(x) вокруг оси y при изменении x от xl до x2.

Эти функции рассчитаны на применение команды **Simplify**, Если решение в аналитическом виде найти невозможно, можно использовать команду **approX** для получения численного решения. Представленный ниже пример поясняет применение некоторых из этих функций:

```
1: "Дополнительные интегральные функции"

                  2
2:      FOURIER (x  , x, 0, 1, 2)

        COS (4 pi x)   SIN (4 pi x)   COS (2 pi x)   SIN (2 pi x~
        ____________________________+______·_______________·______~
3:             2            2 pi            2             pi      ~
          4 pi                             pi                     ~
        ~)      1
        ~— + ——
        ~       3
        ~

                  2
4:      LAPLACE (t  , t, S)

         inf
        /     2   - s t
5:      /    t #e       dt
         0
```

6: $\text{LAPLACE}(t^2, t, 3)$

7: 0.0740740

8: $\text{ARC\_LENGTH}(x^2, x, 1, 2)$

9: 3.16784

10: $\text{SURFACE\_AREA}(x^2 + y^2, x, -1, 1, y, -1, 1)$

11: 7.44625

12: $\text{AREA\_OF\_REVOLUTION}(x^2, x, 1, 2)$

13: **49.4162**

В этом примере возможности функций интегрирования даны выборочно. Тестирование остальных функций можно провести аналогично.

Вспомогательные функции

Функции файлов **english.mth** и **metric.mth**

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.

Функция файла taylor.mth

Начиная с версии Derive 2.60 система содержит файл 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, **x0, y0, v0,** n) — упрощается к ряду Тейлора n-й степени, являющемуся решением уравнения *y»* **= r(x,** y, v). Здесь v обозначает y', а v0 — начальное значение v при x = x0. Например, TAYLOR_ODE2 <2*x*v + x^2*y + 3*x, x, y, v, 0, 0, 1, 5)

упрощается к

$$\frac{3x^5}{10}+\frac{5x^3}{6}+x$$

Вы можете проверить это приближенное решение, подставив его в неупрощенную разность двух частей дифференциального уравнения вместо у и производную от этого выражения вместо **v**, и затем определить, равняется ли ряд Тейлора третьего порядка от этой разности 0. Здесь используется ряд третьей, а не пятой степени потому, что вторая производная понижает степень ряда на 2.

TAYLOR_ODE2 упрощается к ?, если **r** недостаточно дифференцируемо для конечного ряда n-й степени. Если это произошло, то попробуйте положительное целое п. поменьше.

TAYLOR_SOLVE (u, X, y, xO, yO, п) упрощается к ряду Тейлора n-й степени, являющемуся решением у(х) уравнения **u(x,** у) = 0, если известно, что u(x0, yO) = 0. Например, невозможно в общем случае точно решить относительно у трансцендентное уравнение sin у + у + х **#e^x** = 0, а дополнительный нечисловой параметр х делает невозможным общее численное решение. Однако **TAYLOR_SOLVE** раскрывает любое точное решение для конкретного значения **x** в конечный ряд, полезный в окрестности этой точки х, при условии, что такой конечный ряд существует. Например, если yO = 0 при xO = 0 — очевидное решение для одного конкретного значения х, то для получения решения в виде конечного ряда в окрестности этой точки необходимо упростить выражение:

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, yO, п) упрощается к, ряду Тейлора, аппроксимирующему функцию, обратную, у = и(х) в окрестности у = **u(x0)**. Она полезна тогда, когда и(х) не может быть точно инвертирована в замкнутой форме. Например,

TAYLOR_INVERSE (x*#e^x, x, y, 0, 3)

упрощается к

$$\frac{3y^3}{2}-y^2+y$$

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 (см. п. 6.10). **Например,**

POLY_INTERPOLATE ([0,0], [1,1], [2,2], [3,0]], x)

упрощается к

$$\frac{x^2\,(3 - x)}{2}.$$

Демонстрационные файлы (с расширением .dmo)

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 *ч* 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 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 опробовать самостоятельно любой их фрагмент. В систему входят также три демонстрационных файла графики:

plot2d.mth — демонстрационная двухмерная графика в декартовой системе координат;

plotpara.mth — демонстрационная графика для параметрически заданных функций;

plot3d.mth — демонстрационная трехмерная графика.

На рис. **11.3** показана работа с загруженным файлом plot3d. В левом окне виден текст документа — комментарии и математические выражения для построения 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).**

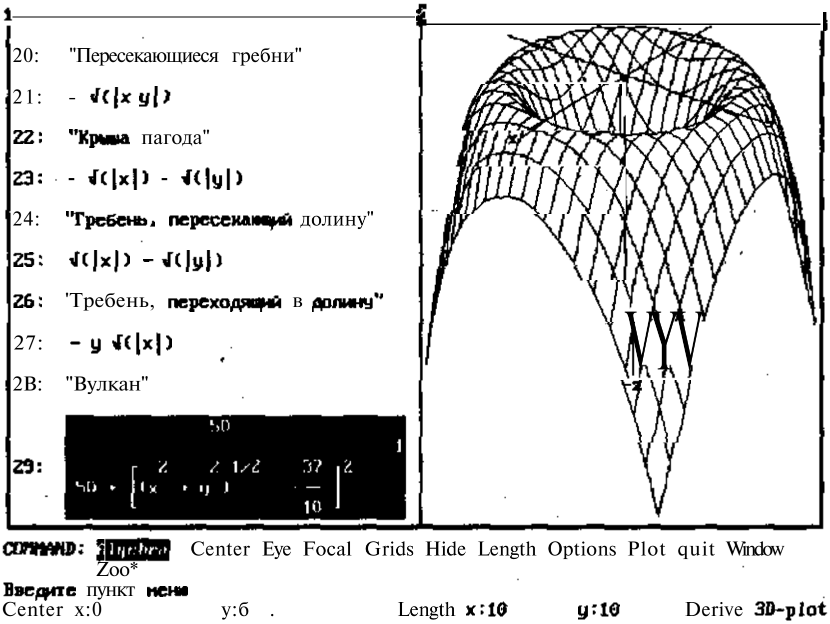


Рис. 11.3. Пример работы с демонстрационным файлом plot3d

С демонстрационными файлами и их работой весьма полезно ознакомиться тем пользователям системы, которые используют ее для графических работ. Каждый файл содержит множество примеров на графические построения, 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 придаются файлы с документацией, подробно описывающие входящие 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 не используемых функций оказывается просто бесполезной.

Чему мы научились:

- Использовать дополнительные векторные и матричные функции.
- Вычислять специальные математические функции.
- Работать с графическими функциями расширения **graphics.mth.**
- Применять рациональную аппроксимацию из расширения approx.mth.
- Использовать функции расширения **misc.mth**
- Применять функции численного дифференцирования и интегрирования
- Работать с демонстрационными файлами.

Урок 12. Расширения для решения уравнений

- Решение нелинейных систем уравнений — расширение файлsolve.mth
- Решение дифференциальных уравнений — расширение файл odel.mth
- Решение ОДУ второго порядка — расширение ode2.mth
- Функции файла ode_appr.mth
- Решение рекуррентных уравнений — расширение requreqn.mth

Решение нелинейных систем уравнений — расширение solve.mth

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:

f1(...x, y, **z)** = 0, ... fn(...x, y, z) = 0.

Для решения таких уравнений известным методом Ньютона — Рафсона можно воспользоваться функцией

NEWTONS(u, x,x0, п).

Здесь и — вектор, содержащий функции **f1, ..., fn;** x — вектор переменных ...x, y, z — аргументов функций; **x0** — вектор начальных значений переменных ...x, y, z; n — число итераций. Функция при ее упрощении возвращает вектор **решения.** Если нужно найти решения, содержащие комплексные числа, то надо использовать эти числа в начальном приближении.

В файле solve.mth функция **NEWTONS** определена следующим образом:

```
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 к_. Знак **«_»** используется, как уже отмечалось, для отличия этой переменной от переменной к, 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, xO, п)

Здесь **g** — вектор левых частей системы нелинейных уравнений, представленных в виде

fl(...x, y, z) = ..., ..., fn(...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 имеют автоматической остановки по достижении погрешностью решения заданного значения. Итерации выполняются заданное 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:

x^2 + y^2 — 5 =O,
x + y -3 = 0.

Диалог с Derive для этого примера имеет вид:

```
1: "Решение системы нелинейных уравнений"

                           / 2      2                   \
2:        NEWTONS (\x   + y   - 5, x + y - 3/, [x,y], [0.8,2.2],5)

           /    0.8        2.2     \
           |                       |
           |  0.971428   2.02857   |
           |                       |
           |  0.999227   2.00077   |
3:         |                       |
           |     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 у для заданных п итераций. Видно, что уже начиная с третьей итерации решение **перестает** меняться, 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 удается 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. В этом случае разумно остановить вычисления нажатием клавиши **Esc.** Попробуйте также для этого случая использовать команду **approX,** если вы до этого применяли **Simplify.** Помните также, что решения в виде матриц помещаются на экране, если п меньше 10.

Решение дифференциальных уравнений (файл odel.mth)

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 имеет три библиотечных файла внешних расширений, ориентированных на решение дифференциальных уравнений различного типа.

Частное решение для **начальных условий y** = yO **при** x = xO

В файле odel.mth содержатся функции для решения различных дифференциальных уравнений первого порядка с применением команды **Simplify.** Ниже перечислены эти функции:

Функция файла odel.mth	**Решение уравнения**
DSOLVE1 (p,q,x,y,x0,y0)	p(x,y)+q(x,y)y'=0 — частное решение;
SEPARABLE **(p,q,x,y,x0,y0)**	y'=p(x)q(y) — частное решение;
LINEAR1_GEN (p,q,x,y,c)	y'+p(x)y=q(x) — общее решение, содержащее константу с;
LINEAR1 (p,q,x,y,x0,y0)	y'+p(x)y=q(x) — частное решение;

	— общее решение (r однородно);
HOMOGENEOUS	— частное решение (r однородно);
EXACT_GEN (p,q,x,y,c)	p(x,y)+q(x,y)y'=0 — общее решение, если оно в полных дифференциалах;
EXACT (p,q,x,y,x0,y0)	p(x,y)+q(x,y)y'=0 — частное решение, если оно в полных дифференциалах;
INTEGRATING_FACTOR (p,q,x,y,x0,y0)	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).

Общее решение, содержащее константу с

Для получения общего решения, содержащего константу с в файле odel.mth, находятся следующие функции:

Функция файла odel.mth	**Общее решение уравнения**
DSOLVE1_GEN(p,q,x,y,c)	p(x,y)+q(x,y)y'=0;
SEPARABLE_GEN (p,q,x,y,c)	y'=p(x)q(y);
LINEAR1_GEN (p,q,x,y,c)	y'+p(x)y=q(x);
HOMOGENEOUS_GEN (r,x,y,c)	y'=r(x,y) если r однородно;
EXACT_GEN (p,q,x,y,c)	p(x,y)+q(x,y)y'=0, если оно в полных дифференциалах;
INTEGRATING_FACTOR_GEN (p,q,x,y,c)	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).

Развитые методы решения ОДУ первого порядка

Следующие функции файла odel.mth поддерживают развитые методы решения обыкновенных **дифференциальных** уравнений (ОДУ) первого порядка:

Функция файла odel.mth	**Решаемое уравнение**
MONOMIAL_TEST (p,q,x,y) интегрирующий множитель уравнения 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 **(r,p,q,k,x,y,x0,y0)**	y'=r(p*x+q*y+k) если p, q, k константы;
LIN_FRAC (r,a,b,c,p,q,k,x,y,x0,y0)	y'=r((ax+by+c)/(px+qy+k));
ALMOST_LIN (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, дается функциями:

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 (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 команды **Solve** для поиска одного или нескольких явных решений.

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, x 0, y 0)

3: $$y = \frac{q}{p} - \frac{q\ \#e^{-p\,x}}{p}$$

4: $$\mathrm{LINEAR1}\ (2\ x,\ x^{2},\ x,\ y,\ 0,\ 0)$$

5: $$y = \#e^{-\mathrm{INT}\ (2\ x,\ x,\ 0,\ x)} \int_{0}^{x} x^{2}\ \#e^{\mathrm{INT}\ (2\ x,x,0,x)}\ dx$$

6: $$\frac{x}{2} + \frac{\#i\ \mathrm{SQRT}\ (pi)\ \#e^{-x}\ \mathrm{ERF}\ (\#i\ x)}{4}$$

7: $$\mathrm{BERNOULLI}\ (2\ x,\ x^{2},\ 0,\ x,\ y,\ 0,\ 0)$$

8: $$y = \#e^{-\mathrm{INT}\ (2\ x,\ x,\ 0,\ x)} \int_{0}^{x} x\ \#e^{2\ \mathrm{INT}\ (2\ x,x,0,x)}\ dx$$

9: $$\mathrm{FUN\_LIN\_CCF}\ (2\ x,\ 1,\ 1,\ 1,\ x,\ y,\ 0,\ 0)$$

10: $$\int_{1}^{x+y+1} \frac{1}{2\ x + 1}\,dy - \int_{0}^{x} 1\ dx = 0$$

11: $$\frac{2\ y - 1}{2\ (2\ x + 1)}\ x + \frac{1}{2} = 0$$

12: $$\mathrm{CLAIRAUT}\ (x^{2},\ x,\ x,\ y,\ 1,\ 1)$$

13: $$\left[\lim_{1 \to 1} (x^{2} - x) = 0,\ \frac{d}{d1}\,x = 0\right]$$

Решение ОДУ второго порядка (файл **ode2.mth**)

Файл ode2.mth содержит ряд функций (включая тестовые со словом TEST в конце имени для проверки допустимости 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 (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 (q,x,y,x0,y0,v0)	**y"=q(y),** У=уО и **y'=v0** при **x=x0;**
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 первого порядка; понижает порядок
EXACT2 (p,q,x,y,v,c)	**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, то возвращается сообщение inapplicable (неприменимо). Применение некоторых из этих функций поясняют следующие примеры:

```
1: "Решение дифференциальных уравнений второго порядка"

2:        LIN2_P0S (a, b, 0, x)

                                                                2        ~
                2                                 - x (SQRT (a   - 4 b)~
             x  (SQRT (a   - 4  b)/2 - a/2)    c2 #e                       ~
3: c1 #e                          ______________________________________   ~
                                                           2               ~
                                             SQRT (a   - 4  b)             ~
```

```
        ~
        ~ + a) / 2
        ~
        ~——————
        ~
        ~

4:      LIN2A_POS (0, a, 0, x)

                                         - #i SQRT (a) x
                #i SQRT (a) x       #i c2 #e
5:      c1 #e                   +——————————————————————
                                        2 SQRT (a)

6:      LIOUVILLE (O, 1, x, y, 1, 1)  .

            y
7:      #e  + x - 1 = O
```

Функции файла ode_appr.mth

Файл ode_appr.mth содержит дополнительные расширения для решения дифференциальных уравнений. Для решения одиночных дифференциальных уравнений вида у'=г(х) при у(хО)=уО в нем заданы функции, представленные ниже:

DIRECTION_FIELD (r,x,x0,xm,m, y,y0,yn,n) — строит сетку с координатами (хО,уО) и (xm,yn) противолежащих углов (используются команды **approX** и **Plot)**;

TAY_ODE1 (r,x,y,x0,y0,n) — решение в виде многочлена Тейлора n-й степени;

PICARD (r,p,x,y,x0,y0) — уточнение приближения по заданному приближению р(х),

EULER (r,x,y,x0,y0,h,n) — дает матрицу [[x0,y0], ... ,[xn,yn]] решения методом Эйлера при шаге h (используется команда **approX)**.

На основе применения этих функций в файле ode_appr.mth определены также и функции для решения систем обыкновенных дифференциальных уравнений, заданных в виде векторов у'=г(х) и вектора начальных условий уО = у(хО):

TAY_ODES (r,x,y,x0,y0,n) — дает решение в виде рядов Тейлора n-ой степени;

PICARD(r,p,x,y,x0,y0) — дает улучшение аппроксимации в виде вектора р(х);

RK (r,v,v0,h,n) — дает решение методом Рунге — Кутта в виде матрицы **[[x0, y0(x0), ym(x0)]**, [хп, уО(хп), ут(хп)]] с шагом решения h (используйте команду **approX)**,

EXTRACT_2_COLUMNS (A,j,k) — выделяет столбцы j, k матрицы A для графического представления результатов решения методом 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))
RK_AUX0(p,v,v0,n):=ITERATES(u_+RK_AUX1(p,v,u_,LIM(p,v,u_)),u_,v0,n)
RK(r,v,v0,h,n):=RK_AUX0(h*APPEND([1],r),v,v0,n)
```

Это определение дает прекрасный пример техники программирования на языке системы Derive и свидетельствует о компактности программ. К примеру, программы решения системы дифференциальных уравнений методом Рунге — Кутта на Бейсике или Паскале, по крайней мере, в несколько раз длиннее, чем приведенная программа из пяти строк. Эта программа является наглядным примером функционального программирования, когда конечная функция определена через ряд промежуточных функций.

Функция **RK** в документах используется в виде

RK(r, v, vO, **h,** n).

В этой функции **r** — вектор левых частей системы обыкновенных дифференциальных уравнений первого порядка вида

y' = fl(...x,y,z) ... z' = fn(...x,y,z).

Вектор v должен содержать перечисление всех переменных .x,y,z, а вектор vO — их начальные значения. Шаг решения по переменной x задается значением h, а число точек решения — значением п. Решение представляется в виде матрицы, столбцы которой дают значения переменных ...x, y, z.

Рассмотрим решение следующей системы дифференциальных уравнений:

y' = z и z' = (y/x - z)/x - y

с начальными условиями: x = 0.3, y = 0.1484 и z = 0.483. При h = 0.1 и п = 7 решение представлено в следующем примере:

1: "Решение системы дифференциальных уравнений"

2: $$\mathrm{RK}\left(\left[z, \frac{\frac{y}{x} - z}{x} - y\right], [x,y,z], [0.3,0.1484,0.483], 0.1, 7\right)$$

$$3: \quad \begin{pmatrix} 0.3 & 0.1484 & 0.483 \\ 0.4 & 0.196090 & 0.470204 \\ 0.5 & 0.242322 & 0.453852 \\ 0.6 & 0.286748 & 0.434113 \\ 0.7 & 0.329038 & 0.411164 \\ 0.8 & 0.368881 & 0.385200 \\ 0.9 & 0.405985 & 0.356439 \\ 1 & 0.440083 & 0.325121 \end{pmatrix}$$

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 численном виде.

Решение рекуррентных уравнений (файл recureqn.mth)

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 — Кутта), поскольку описывающие их рекуррентные формулы нередко имеют простой вид.

Решение рекуррентных уравнений задано в файле recureqn.mth рядом функций:

Функция файла	Решаемое уравнение
LIN1_DIFFERENCE (p,q,x,x0,y0)	решение уравнений **y(x+1)=p(x)y(x)+q(x)** при **y(x0)=y0;**
RECURRENCE1 (r,x,y,x0,y0,n)	п шагов решения **y(x+1)=r(x,y(x))** при y(x0)=y0;
GEOMETRIC1 (k,p,q,x,x0,y0)	решение **y(k*x)=p(x)y(x)+q(x)** при **y(x0)=y0;**
CLAIRAUT_DIF (p,q,d,x,y)	решение p(y-xd)=q(d), где d есть **y(x+1)–y(x);**

LIN2_CCF_DISC (p,q)	**дискриминант d для** y(x+2)+p*y(x+1)+q*y(x)=r(x);
LIN2_CCF_POS (p,q,r,x)	**решениу(x+2)+p*y(x+1)+q*y(x)=r(x) для d>0;**
LIN2_CCF_NEG (p,q,r,x)	**решение y(x+2)+p*y(x+1)+q*y(x)=r(x) для d<0;**
LIN2_CCF_0 (p,q,r,x)	**реш е y(x+2)+p*y(x+1)+q*y(x)=r(x)** для d=0;
IMPOSE_BV2 (x,y,x0,y0,x2,y2,c1,c2)	дает **c1** и c2 для y(x), так **что (x0)=y0 и y(x2)=y2.**

Ниже представлены примеры на решение рекуррентных уравнений:

1: "Решение рекуррентных уравнений"

2: **LIN1_DIFFERENCE** (x^2, x, x, a, b)

3:
```
          2 //x - 1   @1  \             2  \
  (x— 1)! || SUM——————|  (a - 1)!  + b|
          ||@1=a    2 |                 |
          \\     @1!  /                 /
  ————————————————————————————————————————
                                2
                       ( a - 1)!
```

4: CLAIRAUT_DIF (x^2, x, d, x, y)

5: $x^2 - x = 0$

6: **LIN2_CCF_DISC** (x^2, x)

7: $x^4 - 4x$

8: **LIN2_CCF_POS** $(x^2, x, 1, 0)$

9: c1 + c2

Как видно из данного примера, решения могут быть найдены в символьном виде. При этом используются команды основного меню **Simplify** и **approX.**

Чему мы научились:

- Решать нелинейные системы уравнений.
- Решать дифференциальные уравнения первого порядка.
- Решать ОДУ второго порядка.
- Применяит функции файла ode_appr.mth.
- Решать рекуррентные уравнения.

Список литературы

1. *Дьяконов В.П.* Справочник по расчетам на микрокалькуляторах. 3-е изд., доп. и перераб. М.: Наука: Физматлит, 1989. 464 с.
2. *Дьяконов В.П.* Справочник по алгоритмам и программам на языке Бейсик для персональных ЭВМ. М.: Наука: Физматлит, 1987. 240 с.
3. *Дьяконов В.П.* Форт-системы программирования персональных ЭВМ. М.: Наука: Физматлит, 1992. 352 с.
4. *Дьяконов В.П.* Компьютер в быту. Смоленск: Русич, 1996. 640 с.
5. *Дьяконов В.П.* Компьютерная математика. Теория и практика. М.: Нолидж, 2001. 1296 с.
6. *Дьяконов В.* Derive — жемчужина символьной математики // Монитор-Аспект. — 1993. № 2. С. 36.
7. *Дьяконов В., Бирюков С.* Derive в России. Монитор. 1995. № 3. С. 56.
8. *Дьяконов В.П.* Справочник по применению системы Derive. М.: Наука: Физматлит, 1996. 144 с.
9. *Дьяконов В.П.* Справочник по системе символьной математики Derive. М.: СК Пресс, 1998. 256 с.
10. *Дьяконов В., Новиков Ю., Рынков В.* Компьютер для студента; Самоучитель. СПб.: Питер, 2000. 592 с.
11. *Лобанова О.В.* Практикум по решению задач в математической системе Derive. М.: Финансы и статистика, 1999. 540 с.
12. Руководство пользователя DERIVE. Математический Помощник на ПК/ Рус. пер. МНИИТЦ «Скан», Soft Warehouse, Inc., 1993. 348 с.
13. *ДэвенпортДж., Сирэ И., Турнъе Э.* Компьютерная алгебра. Системы и алгоритмы алгебраических вычислений. М.: Мир, 1991. 352 с.
14. *Саймон Б.* Символьная математика: новые времена — новые формы. — PC Magazine. // 1992. № 5.
15. *Гантмахер Ф.* Теория матриц. М.: Наука: Физматлит, 1988. 552 с.
16. Справочник по специальным функциям с формулами, графиками и математическими таблицами/Под ред. М. Абрамовича и И. Стиган. М.:Наука: Физматлит, 1979. 832 с.
17. *Корн Г., Корн Т.* Справочник по математике для научных работников и инженеров. М.: Наука, 1973. 832 с.
18. *Воднев В.Т., Наумович А.Ф., Наумович Н.Ф.* Основные математические формулы. Минск: Вышэйшая школа, 1988. 270 с.
19. *Хювейн Э., Сеппянен И.* Мир Лиспа: В 2 т. М.: Мир, 1990. Т. 1. 447 с; Т. 2. 319 с.

20. *Berry J.S., Graham E., Watkins A.J.* (University of Plymouth). Learning Mathematics through DERIVE. Ellis Horwood, 1993. 371 p.
21. *Josef B.* Teaching Mathematics with DERIVE. Great Britain: Chartwell-Bratt Ltd, 1992. 300 p.
22. *Denton ВЛ.* Learning Linear Algebra Through DERIVE // Ellis Horwood, 1994. 300 p.
23. *Rich AD., Stoutemyer D.R.* Inside the DERIVE Computer Algebra System. England // The International Derive Journal. 1994. Vol.1. № 1. P. 3.

Приложение. Перевод сообщений Derive

Эти сообщения были включены в файл **derive.lan** ряда версий Derive. Их полезно знать и при работе с новыми англоязычными версиями. Поэтому ниже воспроизводится перевод этих сообщений на русский язык:

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 — z координата автоматически
Auto z length — Длина по z автоматически
Axis — Ось

Before — Перед
Bottom — Низ
Bounds — Границы

Cannot do implicit plots — Не могу рисовать неявные функции
Cannot find COMMAND.COM — Не могу найти C0MMAND.COM
Case — Выбор
Change — Сменить
Color — Цвет
Columns — Столбцы
Compute time: — Время счета:
Coordinates — Координаты

Degree — Степень
Digits — Знаков
Dimension — Размер
Direction — Направление
Disk full; delete some files and retry — Диск переполнен; удалите несколько файлов и повторите
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 число столбцов на странице

Enter page lines — Введите число строк на странице

Enter parameter domain — Введите область изменения параметра

Enter path (e.g. C:\DERIVE) — Введите путь (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 число строк на деление

Enter significant digits — Введите число значащих цифр

Enter significant digits to display — Введите число 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

Exponential transformations — Экспоненциальные преобразования

Expression too large to edit — Выражение слишком велико для редактирования

File not found — Файл не найден

Font — Фонт

From — С

Input — Ввод

Left — Левый

Length — Длина

Lines — Строк

Loading expression # — Загружается выражение #

Logarithm transformations — Логарифмические преобразования

Lower limit — Нижний предел

Max — Максимум

Min — Минимум

Mode — Режим

Monitor — Монитор

Must be in graphics mode — Необходим графический режим

Mute warning messages — Звуковой предупреждающий сигнал

Next plot — Следующий график

No finite real samples — Нет конечных действительных точек

No solutions found — Решений не найдено

Operator — Оператор

Order — Порядок

Output — Вывод

Overwrite existing file — Записать поверх существующего файла

Paper — Бумага
Plotting bottom of expression — Строится нижняя часть поверхности
Plotting expression — Строится выражение
Point — Точка
Points — Точки
Press H for help — Для получения справки нажмите латинскую букву H
Press any key to continue — Для продолжения нажмите любую клавишу
Press letter for desired subject — Нажмите букву соответствующую выбранному пункту
Printer — Принтер
Printer not ready — Принтер не готов
Printing expression # — Печатается выражение #
Printing window... — Печатается окно...
Projecting row — Проектируется ряд

Questionable solution — Сомнительное решение
Quit printing expressions — Прервать печать выражений

R — Return to Derive — R — Возврат в Derive
Range — Диапазон
Reading file... — Считывается файл...
Remove hidden lines — Убрать невидимые линии
Resolution (или Reso) — Разрешение
Return for **all** or select 1: — Возврат для всех или выберите 1:
Return for **no** more or select next: — Возврат при отрицании или выберите следующий:
Right — Правый
Rows — Строки

Saving expression # — Записывается выражение #
Saving file... — Пишу в файле...
Scaling — Масштабирование
Select amount of factoring — Выберите размер факторизации
Seleet approach direction — Выберите направление подхода
Select arithmetic mode — Выберите тип арифметики
Select arrow key mode — Выберите режим клавиш-стрелок
Select background — Выберите фон
Select case mode — Выберите чувствительность имен к регистрам
Select character set — Выберите набор символов
Select coordinate system — Выберите систему координат
Select display adapter — Выберите адаптер дисплея
Select display mode — Выберите режим дисплея
Select input mode — Выберите режим ввода
Select monitor type — Выберите тип монитора
Select operator — Выберите операцию
Select orientation — Выберите ориентацию
Select plot mode — Выберите режим построения
Select plot point mode — Выберите режим печати точек
Select plot point size — Выберите размер точек
Select preferred branch for roots — Выберите желаемую ветвь корней
Select print range — Выберите диапазон печати
Select printer type — Выберите тип принтера
Select resolution — Выберите разрешение
Select save range — Выберите блок сохраняемых выражений
Select size — Выберите размер
Select text size — Выберите размер текста
Select value or domain of — Выберите значение или диапазон
Select zoom axis — Выберите изменение масштаба осей
Select zoom direction — Выберите направление изменения масштаба
Set — Установить
Simplifying expression — Упрощается выражение
Size — Размер

Start — Начало
Style — Стиль
Syntax error detected at cursor — Над курсором найдена синтаксическая ошибка

Text — Текст
This version not designed for HP 95LX. — Эта версия не подходит для **HP 95LX.**
Too few real sample points in box — **В** области слишком мало действительных точек
Too many variables — Слишком много переменных

Top — 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

Оглавление

Дьяконов Владимир Павлович

СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ DERIVE

Самоучитель и руководство пользователя

Ответственный за выпуск
В. **Митин**

Верстка
А. **Виноградов**

Обложка
Е. Жбанов

Издательство «СОЛОН-Р»
103001, Москва, а/я 82
Телефоны:
(095) 254-44-10, 252-36-96, 252-25-21
E-mail: Solon-R@coba.ru

Приглашаем к сотрудничеству авторов — специалистов в области компьютерных технологий
E-mail: *Solon-Avtor@coba.ru*

---

000 Издательство «СОЛОН-Р»

ЛР № 066584 от 14.05.99

Москва, ул. Тверская, д. 10, стр. 1, ком. 522

Формат 70x100/16. Объем 20 п. л. Тираж 3000 экз.

АООТ «ПОЛИТЕХ-4»
Зак. 23.