

А.В. Матросов

Основы работы в Maple V Rel. 4

Санкт-Петербург
1999

Введение

Maple является системой аналитических вычислений, предназначенной для облегчения решения не только математических задач, но также инженерных, финансовых, экономических и технических, требующих привлечения разнообразного математического аппарата. Пользователем данной системы может быть любой человек, владеющий математикой в пределах необходимых ему знаний для решения задач проблемной области. Это может быть инженер, занимающийся расчетом конструкций, финансист, следящий за потоками денежных масс на предприятии, математик, использующий Maple для облегчения своей работы, и т.д.

Система Maple работает под управлением операционной системы Windows 95, и ее интерфейс пользователя является стандартным для программ, разработанных для выполнения под управлением этой операционной системы: меню, панели инструментов, рабочая область, возможность загрузки нескольких документов Maple в одном сеансе работы и т.д.

Пользователь с помощью клавиатуры набирает команды (функции) в рабочем поле, которые передаются на обработку основному компоненту системы, называемому ядром системы. Названия, или имена команд и функций соответствуют тем действиям, которые они выполняют, что достаточно удобно как для новичка, так и для специалиста. Любая, введенная пользователем правильная команда Maple, немедленно интерпретируется исполняющей системой, и пользователь сразу же видит результат.

Кроме непосредственного ввода последовательности необходимых команд для решения задачи Maple предоставляет собственный язык программирования, операторы которого похожи на операторы любого языка программирования высокого уровня. Это позволяет создавать собственную последовательность действий для решения конкретной часто выполняемой задачи, и оформить ее в виде процедуры, которую впоследствии можно вызывать в любое время.

Изучив небольшой набор команд и правил, можно приступать к работе в системе Maple. Справочная система поможет в дальнейшем освоении и изучении данной системы аналитических вычислений.

Основной компонент Maple - рабочий лист, в который пользователь заносит не только команды Maple, но и комментарии в виде форматированного текста. Развитая система графики позволяет встраивать в рабочий лист двумерные и трехмерные графические отображения, поясняющие и помогающие решению конкретной поставленной задачи.

Обобщая все сказанное в этом кратком введении в возможности Maple, приведем следующий интересный факт. Когда в 1993 году один из авторов находился в поисках системы аналитических вычислений для решения сложной задачи механики деформируемого твердого тела, ему удалось достать другую систему: МАТНЕМАТИСА, но, естественно, без документации и с очень плохой справочной системой. Он обратился к своему знакомому в Южной Африке с просьбой выслать книгу по этой системе. Ответ был достаточно интересен: «Зачем тебе это надо, бери Maple - и наслаждайся!». Автор так и сделал. Нашел Maple и наслаждается им до сих пор.

Графический интерфейс пользователя

Любое Windows-приложение взаимодействует с пользователем через свой интерфейс, который представляет собой «окно» с расположенными в нем разнообразными стандартными компонентами: меню, панелями инструментов, полями ввода и т.д. Эта глава посвящена описанию основных компонентов интерфейса пользователя системы Maple, которому мы уделим столько времени, сколько необходимо для того, чтобы начать работу. Постепенно по мере необходимости будем постоянно обращаться снова к возможностям, предоставляемым интерфейсом. Наша основная цель - научить решать задачи с помощью Maple, а умение работать с интерфейсом придет по мере накопления опыта работы с системой.

Общий вид

Maple имеет стандартный оконный интерфейс пользователя для программ, работающих под управлением Windows 95/Windows NT. При загрузке программы на экране монитора отображается окно интерфейса, представленное на рис. 1. В нем можно выделить несколько основных компонентов (частей).

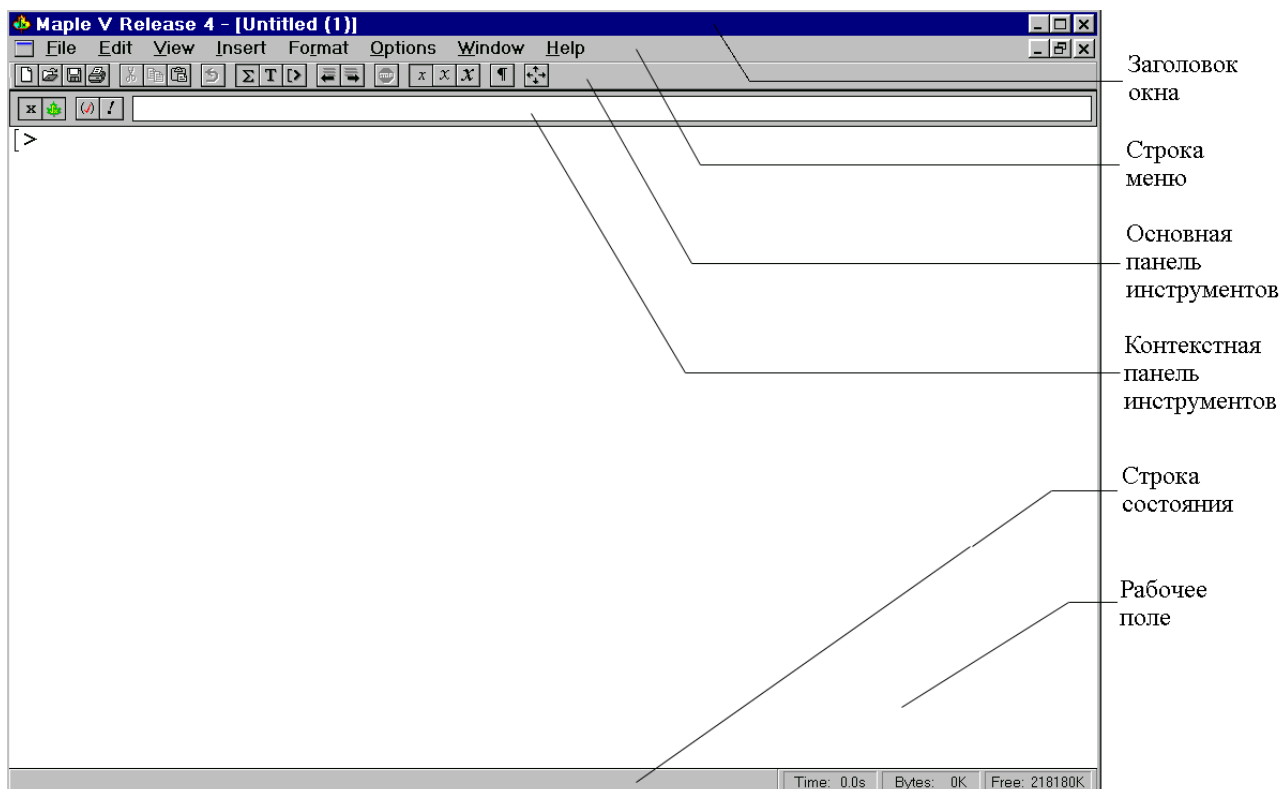


Рис. 1 Окно программы Maple (MF-1.GIF)

В верхней части, непосредственно под заголовком окна, расположена *строка меню*, в которой находятся меню системы Maple. При нажатии левой кнопкой мыши на каком-нибудь из

меню, например **File** (Файл), раскрывается окно, в котором находятся команды и подменю данного меню. В дальнейшем не будем явно указывать, что нажата левая кнопка мыши. Выражение «нажать кнопку мыши» будет соответствовать «нажать *левую* кнопку мыши». Там, где необходимо использовать правую кнопку мыши (например, для вызова контекстного меню), это действие будет указываться явным образом.

Ниже полосы меню расположена *основная панель инструментов* с рядом кнопок, дублирующих наиболее часто используемые команды меню. Щелчок мыши на кнопке приводит к выполнению ассоциированной с ней команды.

Большую часть окна интерфейса занимает *рабочая область*, в которой вводятся команды и отображаются результаты их выполнения.

Непосредственно под основной панелью инструментов расположена *контекстная панель инструментов*, вид которой зависит от того, в какой области рабочего листа расположен курсор. Существует пять видов контекстных панелей инструментов: для выделенного двумерного графика, для выделенного трехмерного графика, для выделенной анимационной графики, для области вывода и для области ввода рабочего листа. На рис. 1 показана контекстная панель инструментов области ввода рабочего листа.

В нижней части интерфейса расположена *строка состояния*, в которой отображаются некоторые параметры, относящиеся к исполняющей системе Maple, а также краткая информация относительно выбранной команды меню.

Во время работы для определенных объектов рабочего листа можно отобразить набор команд, применимых к данным объектам. Для этого достаточно расположить курсор над объектом и нажать правую кнопку мыши. Появится контекстно-зависимое меню для этого объекта (небольшая панель с набором команд).

Рабочие листы

Технология работы в Maple представляет собой интерактивный сеанс: пользователь вводит команды в рабочем поле интерфейса и передает их на выполнение исполняющей системе (ядру) Maple. Все вводимые команды и отображаемые результаты вычислений представляют собой *рабочий лист* - основной документ, который создает Maple и с которым он работает. При завершении сеанса работы его можно сохранить на диске в файлах разных форматов, а при очередном сеансе открыть и снова выполнить все команды, содержащиеся в нем, или произвести его корректировку. На рис. 2 показан рабочий лист с командами и результатами их выполнения в окне интерфейса пользователя системы аналитических вычислений Maple.

Рабочий лист состоит из *области ввода* (Input Region) и *области вывода* (Output Region). В первой пользователь вводит команды Maple, которые передаются на выполнение в ядро системы. В области вывода отображаются результаты выполнения команд и операторов языка Maple, а также двумерная и трехмерная графика, создаваемая графическими командами Maple, если задан режим вставки графики в рабочий лист, а не отображения ее в отдельном окне. В случае отображения графики на рабочем листе область вывода будем называть *областью вывода графики*.

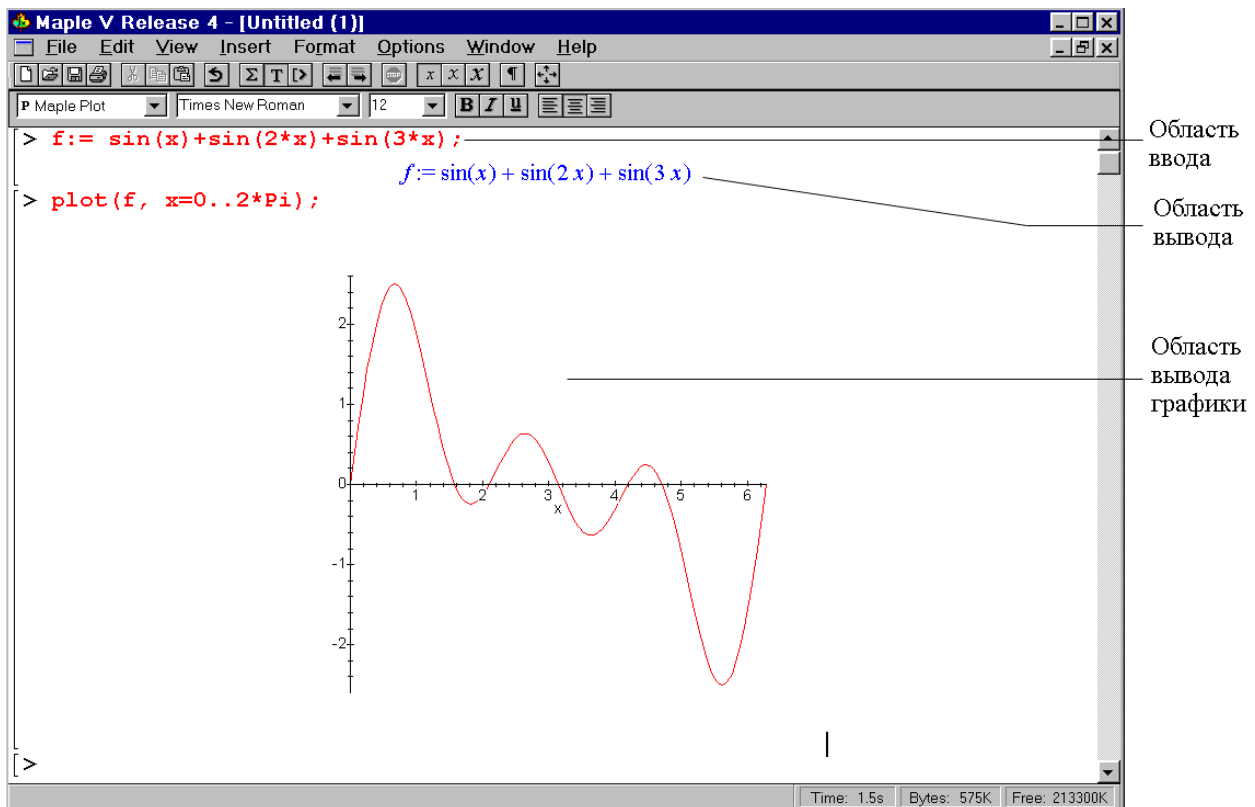


Рис. 2 Рабочий лист в Maple (MF-2.GIF)

Все указанные области можно видеть на рис. 2.

Область ввода и вывода образуют *группу* (Group), которая на рабочем листе отмечается квадратной скобкой. Отметим, что группа может содержать несколько областей ввода и, соответственно, вывода. Основное свойство группы заключается в том, что все ее операторы и команды выполняются за одно обращение к исполняющей системе Maple, т.е. нажатие клавиши <Enter> приводит к вычислению всех ее операторов и команд.

Область ввода

Область ввода - это область рабочего листа, в которой пользователь вводит информацию. Эта информация может быть двоякого рода: команды и операторы Maple или текстовые комментарии.

По умолчанию при создании нового рабочего листа (при загрузке Maple новый рабочий лист создается автоматически) устанавливается режим ввода команд и операторов. Указанием на это является приглашение ввода в строке рабочего листа - символ ' > ', сразу же после которого расположен мерцающий курсор. Пользователь вводит с клавиатуры команды и нажатием клавиши <Enter> передает их на обработку символьному анализатору Maple, который в зависимости от того, правильно или нет они набраны, отображает в поле вывода либо результат выполнения команды, либо сообщение об ошибке.

Команды можно отображать либо в форме операторов Maple, либо в виде привычной математической записи. Например, для вычисления интеграла от функции $f(x)$ можно в поле ввода отобразить команду `int (f (x) , x)`, либо установить режим, при котором эта команда отобразится в привычной математической записи $\int f(x)dx$.

Можно задать режим, при котором в области ввода любая вводимая информация рассматривается как *текстовый комментарий* (Text Region). Такие комментарии удобны, когда необходимо разъяснить, что будет выполнять последующий оператор, или просто необходимо описать постановку задачи и основные моменты алгоритма ее решения. Следует отметить, что система Maple достаточно широко используется по всему миру при обучении математике и некоторым техническим дисциплинам. Поэтому возможность вставки в рабочий лист текстовых комментариев, позволяющих создать учебный материал для изучения и демонстрации применения каких-то понятий, методов, алгоритмов и т.п., является достаточно полезным средством.

Примеры команд в области ввода вместе с текстовыми комментариями можно видеть на рис. 3, из которого видно, что в текстовый комментарий можно вставлять формулы, а также выделять фрагменты текста.

Maple позволяет изменять тип, размер и цвет шрифта как в текстовых комментариях, так и в области ввода. Переключение между режимами ввода команд и текстового комментария осуществляется либо с помощью команд **Text Input** (Ввод текста) и **Maple Input** (Ввод команд) меню **Insert** (Вставить), либо с помощью кнопок на основной панели инструментов (кнопки с символами ' Σ ' и ' T ').

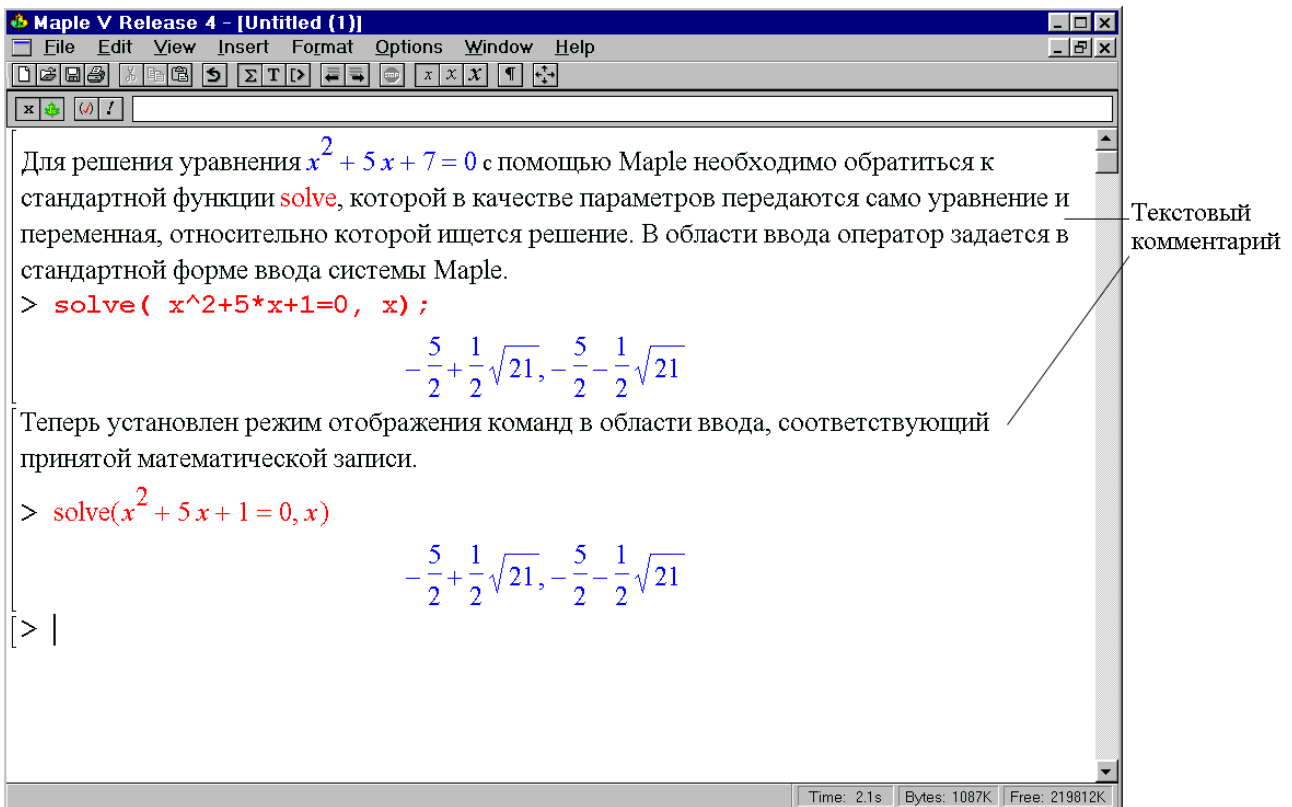


Рис. 3 Рабочий лист с комментариями (MF-3.GIF)

Если оператор не помещается в строке рабочего поля, он автоматически переносится на следующую строку. Пользователю нет необходимости следить за переносом длинных операторов или больших комментариев.

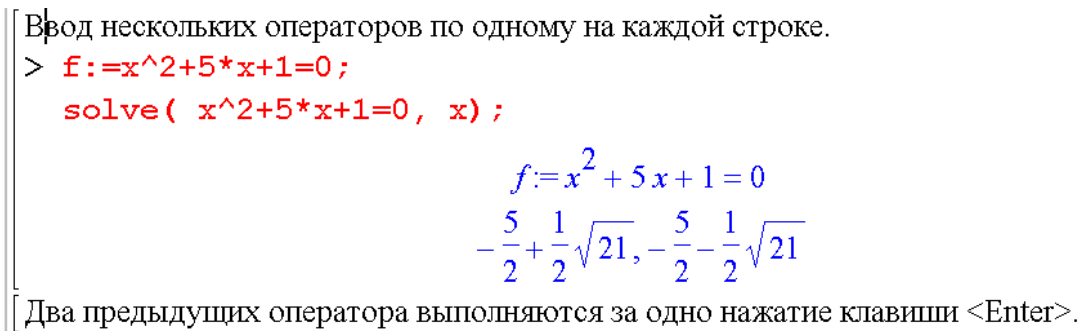


Рис. 4 Несколько операторов, заданных в одной группе (MF-4.GIF)

В одной строке можно вводить несколько операторов, разделенных точкой с запятой (;) или двоеточием (:), но иногда удобно задать несколько операторов по одному на строке, но так, чтобы при нажатии <Enter> они одновременно были переданы символьному анализатору на выполнение. Это осуществляется нажатием комбинации клавиш <Shift>+<Enter> по завершении ввода оператора. В этом случае введенный оператор не пересылается на обработку интерпретатору, а система ожидает ввода пользователя, автоматически переместив курсор на новую строку, причем в новой строке приглашение на ввод не появляется. Таким способом

можно ввести несколько операторов по одному на каждой строке и после нажатия <Enter> все введенные операторы последовательно будут выполнены. На рис. 4 демонстрируется фрагмент рабочего листа с использованной подобной техникой.

Область вывода

Результаты выполнения операторов, введенных в области ввода, отображаются в области вывода. Можно задать три формата вывода: в виде строковой записи, аналогичной формату ввода, в виде символьной записи и в виде типографской записи. Все три типа вывода представлены на рис 5.

Необходимый тип записи в области вывода задается командами **Lineprint Notation** (Строковая запись), **Character Notation** (Символьная запись) и **Typeset Notation** (Типографская запись) подменю **Output Display** (Отображение вывода) меню **Options** (Опции).

```

Вывод в виде строковой записи:
> solve( x^2+5*x+1=0, x );
-5/2+1/2*21^(1/2)  -5/2-1/2*21^(1/2)
>
Вывод в виде символьной записи:
> solve( x^2+5*x+1=0, x );
- 5/2 + 1/2 211/2 , - 5/2 - 1/2 211/2
>
Вывод в виде типографской записи:
> solve( x^2+5*x+1=0, x );
- 5/2 + 1/2 √21, - 5/2 - 1/2 √21

```

Рис. 5 Типы вывода результатов (MF-5.GIF)

Обычно используется вывод в типографском стиле, так как он является наиболее читаемым.

Вывод графики

Как уже говорилось, вывод графики в Maple можно осуществлять непосредственно в рабочий лист (режим по умолчанию) или в отдельное окно. Задание соответствующих режимов осуществляется командами **Inline** (Встроенная графика) и **Window** (Графика в окне) подменю **Plot Display** (Отображение графики) меню **Options** (Опции). Оба типа вывода графики представлены на рис 6.

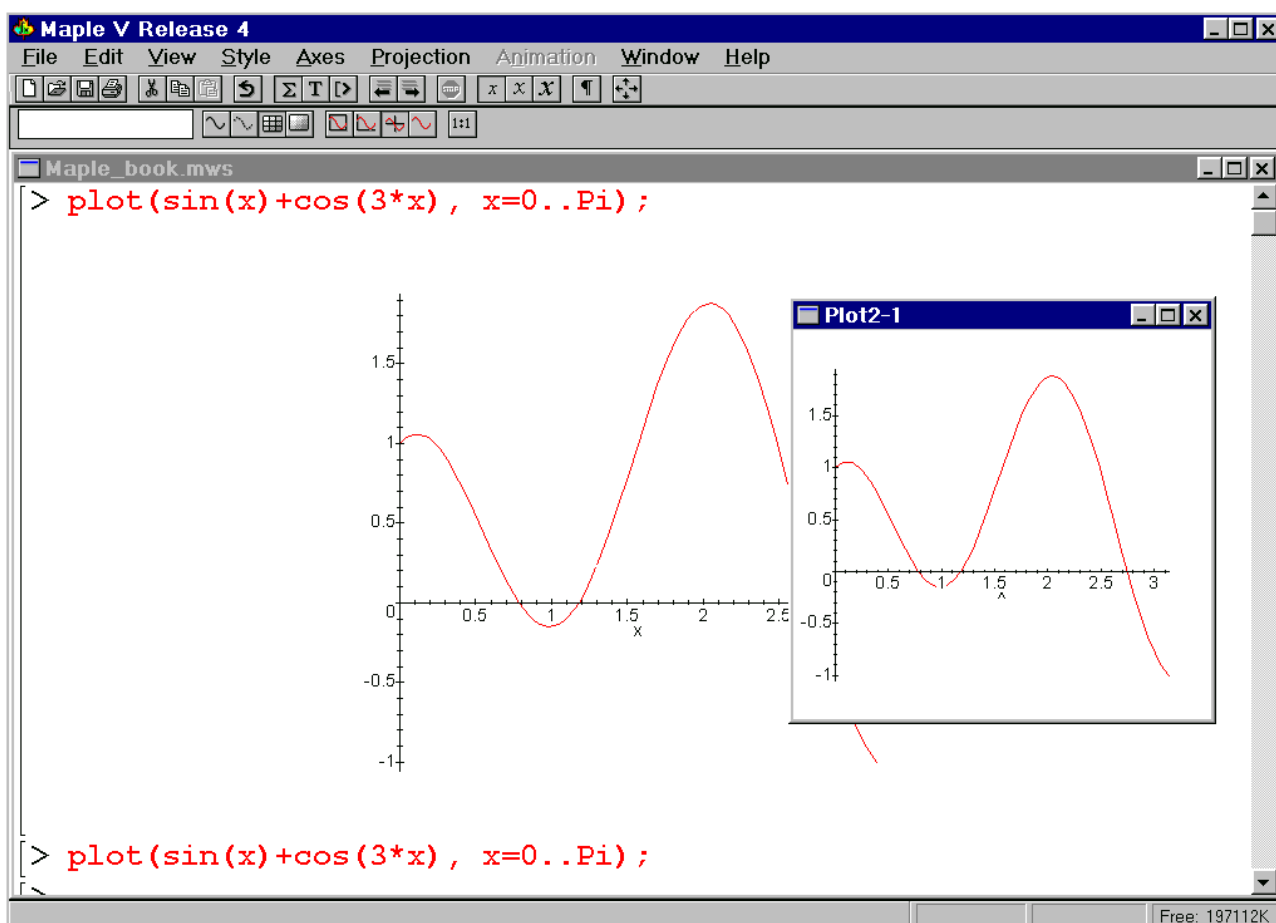


Рис. 6 Два режима вывода графики (MF-6.GIF)

На рисунке показан рабочий лист, в котором открыт документ Maple_book.mws. Два оператора выводят графики одной и той же функции $y = \sin(x) + \cos(3x)$. Отличие заключается в том, что первый оператор `plot` выполняется при включенном режиме вывода графики в рабочий лист, а второй оператор `plot` работает, когда графика выводится в окне, в данном случае в окне plot2-1. Обратим внимание на то, что содержимое полосы меню и контекстной панели инструментов отличаются от соответствующего меню и панели инструментов на предыдущих рисунках. Это связано с тем, что окно графики в данный момент является активным - заголовок окна имеет яркий цвет. Поэтому эти элементы интерфейса пользователя отображают команды, связанные с обработкой графики в окне.

Вывод графики в окне полезен для отображения промежуточных результатов расчетов, так как графические изображения, расположенные на рабочем листе, значительно увеличивают размер файла при сохранении и требуют дополнительной оперативной памяти при открытии документа с графикой.

Работа с меню

Список доступных меню отображается в строке меню. При выборе какого либо меню появляется панель с набором доступных команд и подменю данного меню. Команда отличается от подменю тем, что она сразу же выполняется при щелчке на ней кнопкой мыши, а подменю отображает свой список команд. Выбор какого-либо меню осуществляется щелчком мыши на имени этого меню. На рис. 7 показана панель со списком команд меню **File**.

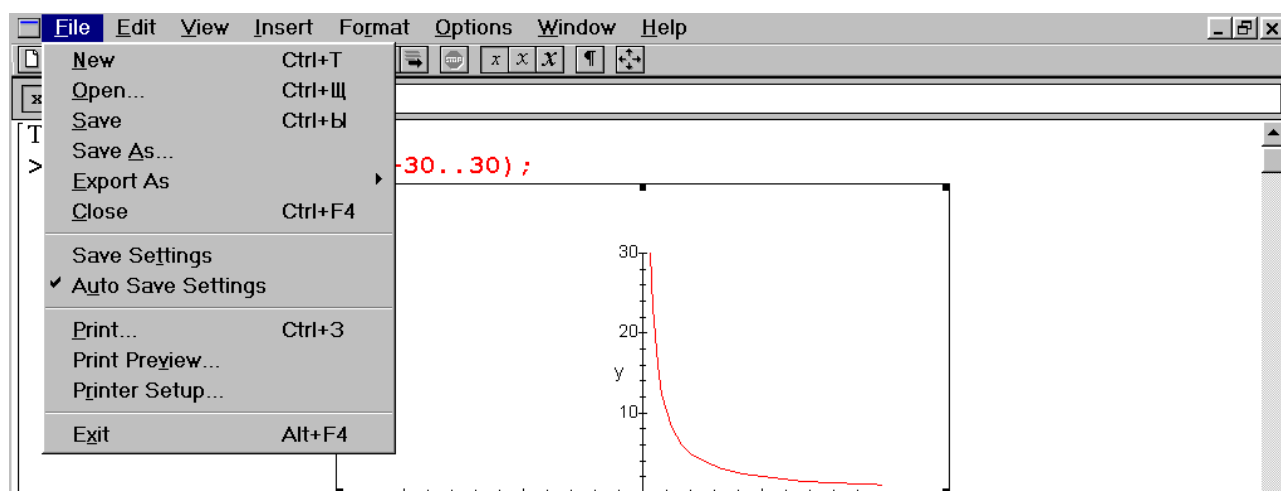


Рис. 7 Команды меню **File** (MF-7.GIF)

На примере меню **File** объясним структуру раскрывающегося списка команд и подменю. Как видно на рис. 7, при выборе меню оно высвечивается (отображается шрифтом белого цвета на синем фоне) и при этом появляется панель со списком команд и подменю.

В этой панели в левом столбце расположены имена (названия) команд, а справа расположены соответствующие каждой команде «быстрые клавиши» - комбинации клавиш, одновременное нажатие которых приводит к выполнению команды. Например, если на клавиатуре нажать одновременно клавиши <Ctrl> и <T> (такие действия принято обозначать <Ctrl>+<T>), то выполнится команда **New** меню **File**. Обратим внимание, что команды могут и не иметь быстрых клавиш, например команда **Save As**.

Если в правом столбце напротив имени команды отображается символ '▶', это означает, что данная команда является подменю, и при ее выборе отобразится дополнительная панель с командами (а может быть и подменю). Команды подменю **Export As** (Записать как) показаны на рис. 8.

Если в названии команды присутствует многоточие, это означает, что для выполнения данной команды требуется дополнительная информация, которую пользователь должен ввести в отображаемом диалоговом окне. Например, для команды **Save As** (Сохранить как) необхо-

можно в диалоговом окне задать имя файла и его расположение (путь). Это можно сделать в диалоговом окне, отображаемом данной командой.

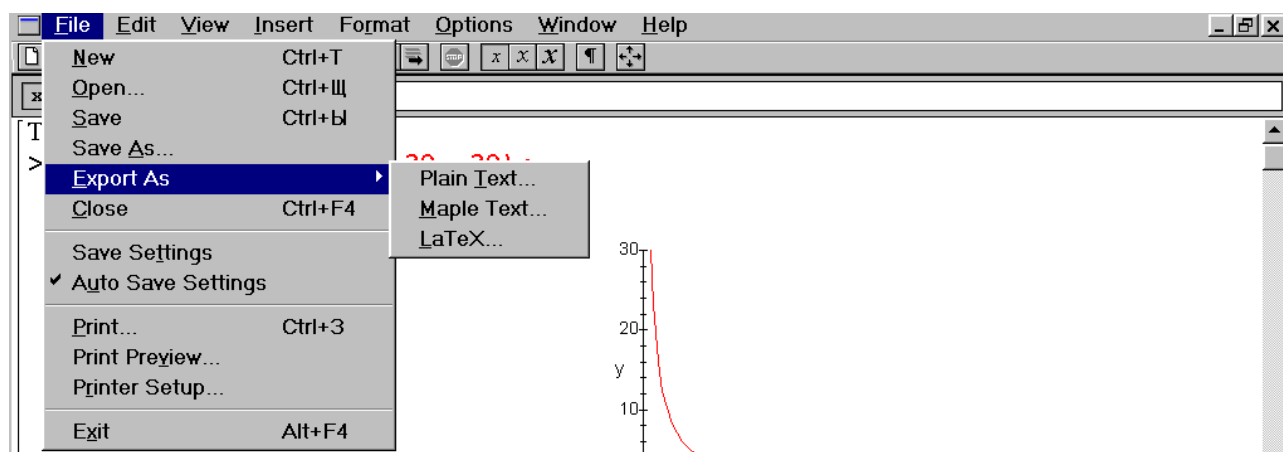


Рис. 8 Команды подменю **Export As** (MF-8.GIF)

Обычно список команд разделен на логические группы, содержащие команды, выполняющие функционально-связанные действия. Такие группы отделяются в панели команд разделительной чертой. Например, команды меню **File**, связанные с печатью, выделены в одну группу.

Действия некоторых команд заключаются в установлении какого-либо режима работы системы Maple. Обычно они действуют как переключатели: включают или выключают соответствующий режим. Если режим, определенный командой, включен, то слева от ее имени отображается метка '✓'. Например, в меню **File** на рис. 8 включен режим **Auto Save Settings** (Автоматическое сохранение установок).

Выбор меню и их команд можно производить и с помощью клавиатуры. Если посмотреть на названия меню, то можно заметить, что некоторые символы отображаются подчеркнутыми. Эти символы используются в качестве быстрых клавиш для выбора меню. При нажатии комбинации клавиш <Alt>+<подчеркнутый символ> отобразится список команд меню, в имени которого соответствующий символ подчеркнут. Например, для меню **File** таким символом является «F» (регистр клавиатуры не имеет значения).

Для выбора команд и подменю из списка меню можно применять «горячие клавиши». Они отличаются от быстрых клавиш тем, что простое нажатие на клавиатуре клавиши, соответствующей подчеркнутому символу в имени команды или подменю при раскрытом списке команд меню, приводит к выполнению соответствующей команды или отображению списка команд подменю. Например, нажатие клавиши <N> при раскрытом списке команд меню **File** приведет к выполнению команды **New** (Новый).

Если в списке команд раскрывающегося меню название команды отображается серым цветом, то данная команда не доступна. Обычно в Maple такое происходит с командами вставки содержимого Буфера обмена или копирования в Буфер обмена. Если он пуст, то, естественно, ничего нельзя вставить в документ, и поэтому команды вставки из Буфера обмена не доступны. В случае если в рабочем листе нет выделенного фрагмента, то наоборот, ничего нельзя скопировать в Буфер обмена, и соответствующие команды недоступны.

Виды меню

Интерфейс системы Maple является интерфейсом, реализующим работу с несколькими документами одновременно. Это означает, что в одно сеансе можно открыть несколько документов Maple и с помощью команд меню **Window** переключаться между ними или отобразить все окна открытых документов.

Концепция интерфейсов, поддерживающих работу со многими документами, так называемые *MDI-интерфейсы* (от английского Multiple Document Interface), предполагает, что при работе с разнородными документами список меню в строке меню может изменяться, чтобы предоставить пользователю команды, специфичные для того типа документа, с которым в данный момент он работает. В Maple существует два типа документов: рабочий лист и графический вывод (если задан режим вывода графики в отдельном окне). С каждым из них связаны свои списки меню, которые отображаются на строке меню при активизации соответствующего документа. Будем называть меню рабочего листа основным, а меню для графики - графическим.

Если графика вставлена в рабочий лист, то при помещении курсора в область графического вывода также отображается графическое меню, позволяющее выполнить ряд команд редактирования графического изображения. Отметим, что графические меню для двумерной и трехмерной графики отличаются, хотя некоторые название меню совпадают, но списки команд в них разные.

Подробно графические меню будут рассмотрены в разделе, посвященном выводу графической информации. В данном разделе кратко остановимся на командах основного меню.

Основное меню

Основное меню отображается при работе с рабочими листами и состоит из следующих меню: **File** (Файл), **Edit** (Правка), **View** (Вид), **Insert** (Вставка), **Format** (Формат), **Options** (Опции), **Window** (Окно) и **Help** (Справка). Здесь кратко опишем основные, наиболее часто используемые при работе с Maple команды этих меню.

Команды меню File

Команды этого меню предназначены для создания, открытия, сохранения и печати рабочих листов.

Новый рабочий лист создается при помощи команды **New** (Новый). Отметим, что при запуске Maple автоматически создает новый рабочий лист. Для сохранения рабочего листа в файле существуют две команды: **Save** (Сохранить) и **Save As** (Сохранить как). Первая сохраняет рабочий лист в том же самом файле, из которого рабочий лист и был открыт, а вторая в файле с другим именем. Команда **Save As** отображает стандартное диалоговое окно системы Windows для задания имени файла и его расположения на жестком диске.

Открыть сохраненный ранее в файле рабочий лист можно командой **Open** (Открыть). При этом в отображаемом диалоговом окне следует задать имя и расположение файла. Команда **Close** (Закреть) закрывает текущий активный рабочий лист, спрашивая пользователя о подтверждении сохранения файла перед закрытием, если в рабочий лист были внесены какие-то изменения.

Рабочие листы сохраняются в файлах с расширением MWS. Существует возможность сохранить рабочий лист в файлах других форматов. Для этого необходимо в диалоге команды **Save As** указать соответствующий формат, либо воспользоваться командами подменю **Export As** (Экспортировать в формате). Мы не будем описывать эти другие форматы, только отметим, что система Maple позволяет сохранять рабочий лист в формате издательской системы LaTeX, достаточно широко используемой для написания математических текстов.

В этом же меню существуют команды печати **Print** (Печать) и предварительного просмотра **Print Preview** (Предварительный просмотр печати) рабочего листа, а также команда **Printer Setup** (Установки принтера) для установки параметров принтера.

Команды меню Edit

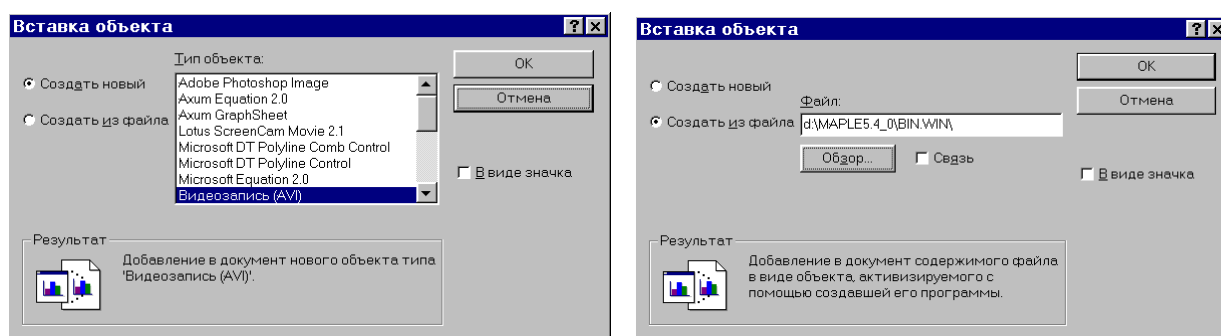
Команды этого меню позволяют редактировать и вносить правку в рабочий лист. В основном они предназначены для работы с текстовыми комментариями, чтобы придать им привлекательный и удобочитаемый вид.

Большинство команд этого меню аналогичны командам меню **Edit** (Правка) других приложений Windows. Команды **Cut** (Вырезать), **Copy** (Копировать) и **Paste** (Вставить) позволяют соответственно вырезать или скопировать выделенный текст в Буфер обмена, а также вставить его содержимое в рабочий лист в месте расположения курсора.

Команда **Undo Delete** (Отменить удаление) позволяет отменить последнее удаление, произведенное на рабочем листе.

Команда **Select All** (Выбрать весь лист) позволяет выделить все содержимое рабочего листа, например, для копирования в Буфер обмена.

Группа команд меню **Edit** позволяет вставить в рабочий лист объекты и документы, разработанные в других приложениях Windows 95. Это называется технологией ActiveX (ранее называвшейся OLE). Команда **Insert OLE Object** (Вставить OLE-объект) позволяет вставить в рабочий лист объект другого приложения. Эта команда вызывает стандартный диалог **Вставка объекта**, позволяющий задать приложение (список **Тип объекта**), объект которого необходимо вставить. При этом можно вставить объект, сохраненный в файле, или создать новый, запустив соответствующее приложение. (Рисунки MF-8a.GIF и MF-8b.GIF)



Для вставки объекта из файла в диалоговом окне следует установить переключатель **Создать из файла** и в поле ввода **Файл**, появившемся в диалоговом окне после нажатия кнопки **ОК**, задать полное имя файла, либо, нажав кнопку **Обзор**, указать нужный файл.

Для создания нового объекта следует установить переключатель **Создать новый** и выбрать нужное приложение из списка **Тип объекта** диалогового окна.

Остальные команды этого меню будут рассматриваться по мере их использования в примерах данной книги.

Команды меню View

Команды этого меню позволяют пользователю задать отображаемые элементы интерфейса пользователя системы Maple, а также отображение информации в рабочем листе.

Команды первой группы из списка команд являются простыми переключателями. Они включают или выключают режим отображения панели инструментов (команда **Tool Bar**), контекстной панели инструментов (команда **Context Bar**) и строки состояния (команда **Status Line**). Если соответствующий элемент интерфейса отображается, то, как указывалось ранее, слева от имени команды стоит «галочка». Команда **Show Invisible Characters** (Показать не-

печатаемые символы) включает или выключает режим отображения непечатаемых символов на рабочем листе. Такими символами являются пробел и символ завершения строки ' ¶ '. Пробел при таком режиме отображается в виде точки, расположенной выше базовой линии строки (чтобы не спутать с точкой - знаком препинания, расположенной на базовой линии строки).

Команды подменю **Zoom Factor** (Изменить масштаб) позволяют изменять масштаб отображения рабочих листов: увеличивать или уменьшать размеры рабочих листов в окне системы Maple.

Команды подменю **Bookmarks** (Закладки) позволяют установить закладку в рабочем листе и быстро переходить к месту, отмеченному определенной закладкой. Это средство удобно для работы с большими документами, а также для отметки, например, текста, который требует редактирования и доработки.

Меню Insert, Format, Options и Window

Команды меню **Insert** полностью посвящены работе с рабочими листами и будут описаны тогда, когда пойдет разговор о математических вычислениях с помощью системы Maple.

Меню **Format** связано с заданием разнообразных стилей вывода информации. Для новичков это не очень существенно, так как обыкновенно начинающего пользователя вполне устраивают системные установки стилей отображения текста в области ввода и вывода. Поэтому вопросы работы с командами этого меню отнесены в раздел дополнительных возможностей Maple.

Команды установки режима вывода результатов команд и операторов Maple в области вывода, а также установки режимов встроенной в рабочий лист и оконной графики рассмотрены ранее. Остальные команды будут освещены по мере необходимости в тексте.

Команды меню **Window** предназначены для работы с несколькими открытыми документами Maple. Они позволяют задать расположение окон документов в окне интерфейса Maple. Их можно расположить каскадом (команда **Cascade**, рис. 9 а)), плиткой (команда **Tile**, рис. 9 б)), горизонтально (команда **Horizontal**, рис. 9 в)) и вертикально (команда **Vertical**, рис. 9 г)).

С помощью команды **Arrange Icons** (Выстроить значки) упорядочиваются значки свернутых окон документов.

Две команды позволяют закрыть все открытые окна (**Close All**) или же закрыть окна, содержащие справочную информацию (**Close All Help**).

Кроме этих команд в меню **Window** отображаются все открытые в данной сессии документы, что позволяет легко и просто переходить от одного документа к другому.

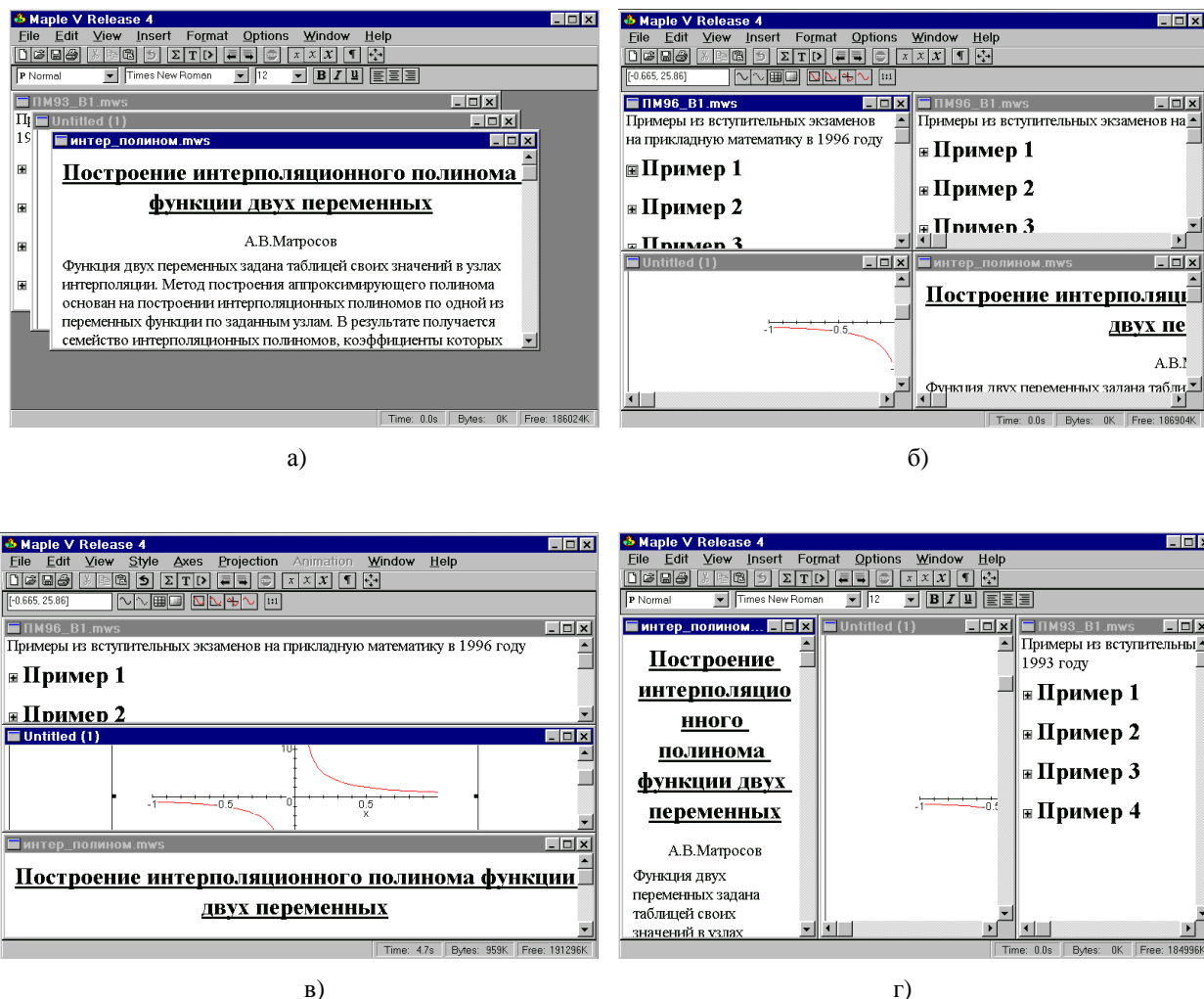


Рис. 9 Различные способы расположения документов в окне интерфейса Maple (MF-9a,b,c,d.GIF)

Справочная система

Последним меню в списке является меню **Help** (Справка). Справка очень полезное и удобное средство. Она позволяет достаточно быстро найти нужную информацию, не прибегая к разнообразным книгам и пособиям, тем более, что в них обыкновенно отражена не вся информация о системе. Поэтому остановимся подробнее на получении справочной информации в системе аналитических вычислений Maple.

В Maple V версии 4.0 справочная система отличается от аналогичной в других приложениях, работающих под управлением системы Windows. Эта система организована в виде гипертекстовых документов.

Организация справки

В основу справочной системы положено понятие гипертекста, т.е. документа, в котором имеются ссылки на другой документ. Таким образом, продвигаясь по этим ссылкам можно получать дополнительную информацию, уточняющую или дополняющую информацию, содержащуюся в исходном документе.

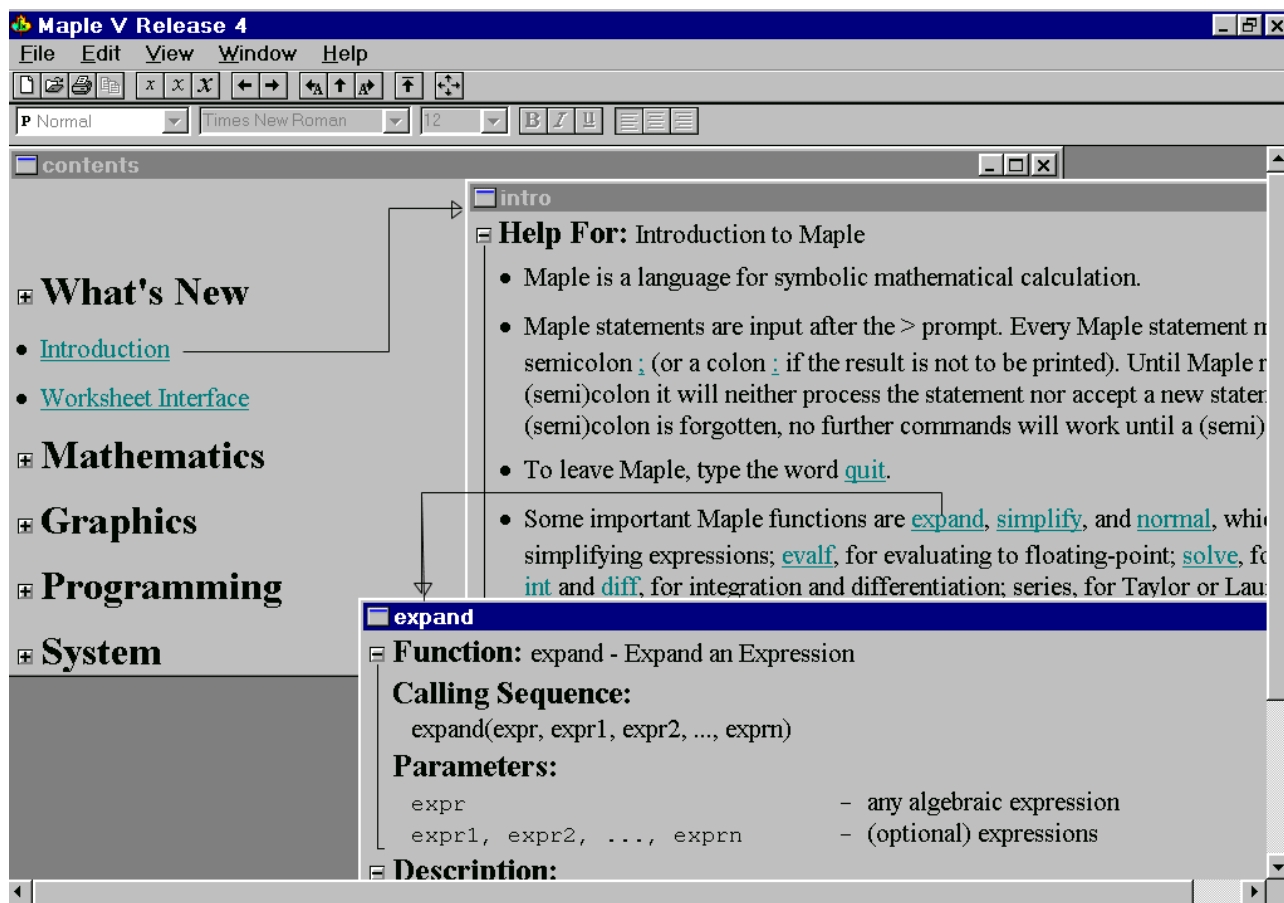


Рис. 10 Гиперсвязи между документами Справки (MF-10.GIF)

Это очень похоже на чтение книги, в которой имеются ссылки на другие главы, разделы или формулы той же книги, или даже на другую книгу. Только при чтении книги необходимо ее перелистывать и обращаться, например, к странице, на которую ссылаются в тексте, или вообще брать в библиотеке другую книгу.

В случае гипертекстовых документов не надо идти в библиотеку, а достаточно щелкнуть кнопкой мыши на соответствующей связи, и необходимый документ отобразится в окне интерфейса Maple.

Такие слова-ссылки (гиперссылки) в гипертекстовых документах отображаются другим цветом и могут быть подчеркнуты. В системе Справки Maple эти ссылки отображаются бирюзовым цветом и подчеркнуты. Если установить курсор мыши на такую связь и щелкнуть кноп-

кой мыши, то загрузится документ справочной системы, с которым установлена связь через слово-ссылку. На рис. 10 показаны три документа справочной системы Maple, которые связаны посредством гиперссылок. Стрелки указывают, какое слово в документе ссылается на следующий.

С помощью гиперссылок можно «путешествовать» по справочной системе Maple, добираясь до нужной информации. Такие «путешествия» полезны еще и тем, что можно натолкнуться на описание каких-то возможностей, с которыми пользователь еще не сталкивался, но которые могут оказаться полезными для него при решении задач проблемной области, в которой он работает.

Вызов справки с помощью меню Help

Команда **Contents** (Оглавление) загружает в рабочее окно документ, содержащий оглавление справочной системы Maple. При этом изменяется список меню в полосе меню, панель инструментов, а контекстная панель инструментов становится недоступной. Документ Справки внешне представляет собой рабочий лист, который нельзя модифицировать, но который содержит гиперсвязи, позволяющие загружать другие документы Справки. Панель инструментов справочной системы содержит кнопки, ассоциированные с командами справочной системы. На рис. 11 показана эта панель инструментов с описанием назначения кнопок.

В меню **Edit** справочной системы есть очень полезная команда - **Copy Examples**. Практически на каждой странице справки есть несколько примеров на использование описываемых функций. Эта команда позволяет скопировать в Буфер обмена примеры, вставить их в рабочий лист и выполнить. Мы настоятельно рекомендуем проделывать эту процедуру для каждой, вновь изучаемой функции и анализировать получаемые результаты.

Команда **Topic Search** (Поиск раздела) меню **Help** позволяет найти необходимый раздел справки, а команда **Full Text Search** (Поиск текста) позволяет найти все страницы Справки, в которых найден заданный в диалоговом окне этой команды текст.

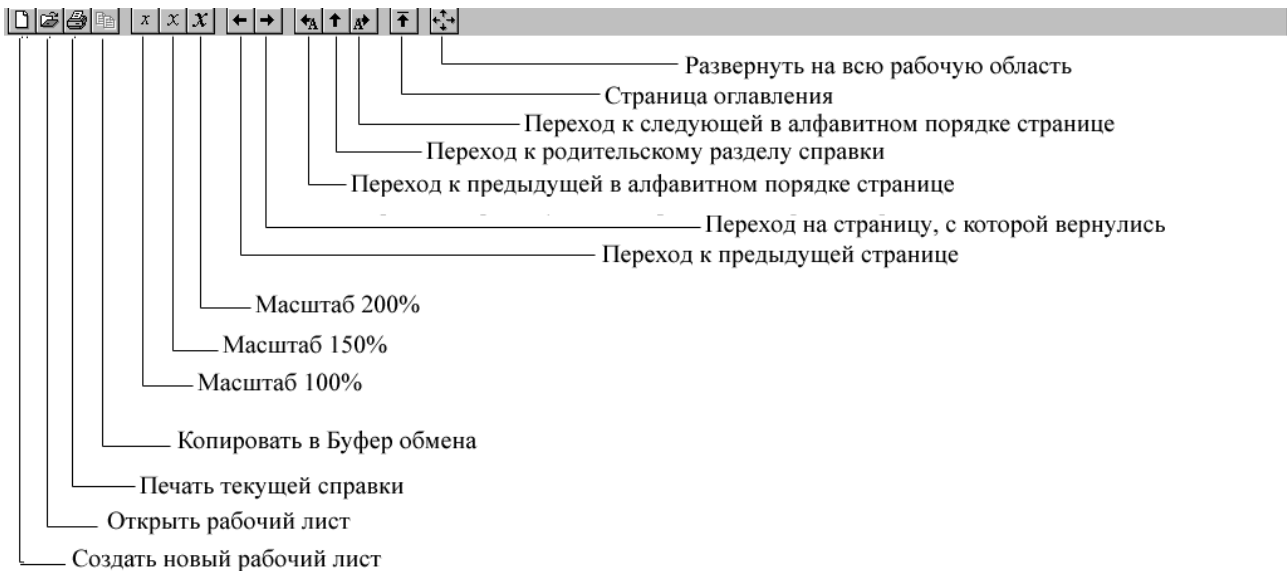


Рис. 11 Панель инструментов справочной системы Maple (MF-11.GIF)

Полезная команда **History** (История) отображает все просмотренные в данном сеансе страницы Справки и позволяет перейти на любую из них.

При работе со Справкой может быть открыто много справочных страниц. Чтобы закрыть все окна страниц Справки, необходимо в меню **Window** выбрать команду **Close All Help** (Закрыть все страницы Справки).

Вызов справки из рабочего листа

Получать справочную информацию можно и не используя команды меню **Help**, набрав команду `?topic`, где `topic` означает имя пакета (библиотеки), команды или служебного слова Maple. (Заметим, что эта команда не завершается точкой с запятой или двоеточием.) Для получения информации о команде `subtopic` из пакета `topic` можно воспользоваться следующими эквивалентными формами команды получения справки: `?topic,subtopic;` и `?topic[subtopic];`. Обратим внимание на то, что эти команды завершаются разделителем (`;`).

Вместо команд со знаком вопроса в начале можно использовать следующие формы команды `help`, завершающиеся точкой с запятой:

```
help(topic);
help(topic, subtopic);
help(topic[subtopic]);
```

Все перечисленные вызовы справки требуют задания команд в области ввода, что может оказаться неудобным. Система Maple предоставляет более удобный способ получения справки

по команде, заданной в рабочем листе. Для этого необходимо поместить курсор на имени команды или служебного слова, и нажать клавишу <F1>. Такой способ, конечно, удобен, если необходимо быстро получить информацию, например, о параметрах нужной команды или об альтернативных формах ее задания. При этом, естественно, необходимо знать имя команды, поэтому предыдущие способы или обращение к командам меню **Help** остаются единственными способами для определения, например, всех команд какого-нибудь пакета.



Основы работы

Система аналитических вычислений Maple является интерактивной системой. Это означает, что пользователь вводит команды в рабочем листе и, нажав клавишу <Enter>, сразу же передает ее аналитическому анализатору системы, который выполняет ее. Если команда введена правильно, то в области вывода появляется результат ее выполнения, если она содержит синтаксические ошибки, система печатает сообщение об этом. Для исправления ошибки следует вернуться к оператору, откорректировать его и снова выполнить. После выполнения введенной команды система ожидает очередной команды от пользователя.

Каждый оператор или команда должны *обязательно* завершаться разделительным знаком. Таких в Maple два - точка с запятой (;) и двоеточие (:). Если предложение завершается точкой с запятой, то оно вычисляется, а в области вывода отображается результат. При использовании двоеточия в качестве разделителя команда выполняется, но результаты ее работы не отображаются в области вывода рабочего листа. Это удобно при программировании в Maple, когда нет необходимости в выводе каких-то промежуточных результатов, получаемых из операторов цикла, так как вывод этих результатов занял бы много места на рабочем листе и время выполнения существенно увеличилось бы.

Объекты, переменные и выражения

Как и в любой интерактивной системе, в Maple реализован свой язык, с помощью которого происходит общение пользователя с системой. Базовыми понятиями являются объекты и переменные.

Простейшими *объектами*, с которыми может работать Maple, являются числа, константы и строки. Если читатель знаком с каким-либо языком программирования, то эти объекты соответствуют литеральным (буквальным) константам разных типов данных в языках программирования.

Числа могут быть целыми, обыкновенными дробями, радикалами и числами с плавающей точкой. Первые три типа чисел позволяют выполнять точные вычисления, без округлений, разнообразных математических выражений. Числа с плавающей точкой являются приближенными, в которых ограничено число значащих цифр.

Для задания дробей используется символ деления (/), а радикалы задаются с помощью символа возведения в степень (^) или (**) и функции sqrt(), определяющей квадратный корень из

числа или выражения, заданного в круглых скобках после имени этой функции. На рис. 12 показаны способы задания чисел и как они обрабатываются интерпретатором Maple.

Целые числа задаются в виде последовательности цифр от 0 до 9. Отрицательные числа задаются со знаком минус (-) перед числом. *Дробь* задается с помощью операции деления двух *целых* чисел. Заметим, что Maple автоматически производит операцию сокращения дробей. *Радикалы* задаются как целые или дробные числа, возведенные в дробную степень, или с помощью функции `sqrt()` с целым или дробным параметром. Здесь также производятся возможные упрощения, что видно из последних двух примеров задания радикалов.

Вычисления с целыми, дробями и радикалами являются абсолютно точными, так как при работе с этими типами чисел Maple не происходит никаких округлений в отличие чисел с плавающей точкой.

```
Целые числа:
> 4; -234;

                               4
                               -234

Дробные числа:
> 4/5; -40/24;

                               4
                               5
                               -5
                               3

Радикалы:
> sqrt(24); (23/2)^(1/2); (23/2)^(3/5);

                               2√6
                               1/2 √23 √2
                               1/2 233/5 22/5
```

Рис. 12 Целые числа, дроби и радикалы (MF-12.GIF)

Числа с плавающей точкой задаются в виде целой и дробной частей, разделенных десятичной точкой, и предшествующим знаком числа, например 2.3456, -3.415. Можно задать целый показатель степени после букв e или E, в которую необходимо возвести число 10 и потом умножить его на заданную мантиссу числа. Например, 2.345e4 соответствует числу 23450.0. Таким образом можно представлять очень большие по абсолютному значению числа (показатель степени положительное число) или очень маленькие (показатель степени отрицательное число).

Из чисел можно составлять математические выражения с помощью арифметических операций.

Символами арифметических операций являются:

Символ	Операция
+	Сложение
-	Вычитание
*	Умножение
/	Деление
^ или **	Возведение в степень
!	Факториал (только целое)

Последовательность выполнения арифметических операций соответствует стандартным правилам старшинства операций в математике: сначала производится возведение в степень, затем умножение и деление, а в конце - сложение и вычитание. Все действия выполняются слева направо. Операция вычисления факториала имеет наибольший приоритет. Для изменения последовательности арифметических операций следует использовать круглые скобки.

Если в выражении все числа являются целыми, дробями или радикалами, то результат представляется также с использованием этих типов данных, но если в выражении присутствует число с плавающей точкой, то там, где это возможно, Maple будет производить вычисления с абсолютной точностью, но некоторые подвыражения будут вычислены с плавающей точкой, как показано на рис.13.

Вычисления с плавающей точкой:

```
> 4.0+sqrt(2/3)*3.0^(1/3)/sqrt(2);
4.0 + .2403749169√6√2
```

Рис. 13 Вычисления с плавающей точкой (MF-13.GIF)

Скажем так, Maple всегда пытается произвести вычисления с абсолютной точностью. Если это не получается, тогда он подключает арифметику с вещественными числами.

Кроме чисел, задаваемых пользователем, Maple содержит целый ряд предопределенных именованных констант – констант, к значению которых можно обращаться с помощью некоторого имени. Часть этих констант не может быть изменена, а часть можно изменять.

К неизменяемым константам относятся следующие:

Константа	Значение
false	Значение «ложь» при работе с булевыми переменными
true	Значение «истина» при работе с булевыми переменными
gamma	Константа Эйлера $\gamma = \lim_{n \rightarrow \infty} \left(\sum_{i=1}^n 1/i - \ln(n) \right) \approx$ $\approx 0.5772156649\dots$
Pi	Число $\pi \approx 3.14159265\dots$
I	Мнимая единица $\sqrt{-1}$
infinity	Бесконечность ∞

Константы, значения которых могут быть переопределены, – это константы, задающие необходимые для работы системы значения. К наиболее важным можно отнести две константы, позволяющие влиять на точность вычислений: `Digits` и `Order`. Первая константа задает число значащих цифр для операций с числами с плавающей точкой. По умолчанию она имеет значение 10. Константа `Order` определяет количество членов в разложении функции в ряд (по умолчанию установлена равной 6).

Кроме чисел Maple позволяет работать со *строкой* - любым набором символов, заключенным в обратные кавычки, например, ``Это пример строки в Maple``. Для работы со строками и строковыми переменными существует набор функций, с которым можно ознакомиться по справочной системе.

Мы изучили, как работать в Maple с числовыми выражениями, но это только позволяет использовать его как некоторый достаточно «умный» калькулятор, не включая всю мощь аналитических вычислений. Первым шагом к освоению всех возможностей Maple является знакомство с переменными, в которых можно хранить вычисленные значения, функции и целые выражения.

Каждая *переменная* Maple имеет имя – последовательность символов, начинающаяся с буквы, причем строчные и прописные буквы считаются различными. Про такие системы гово-

рят, что они чувствительны к регистру. Кроме букв в именах переменных могут использоваться также цифры и знак подчеркивания. Примеры различных имен:

MyName, myname, my_name

В качестве имен запрещено использовать зарезервированные слова языка Maple:

and	by	do	done	elif
else	end	fi	for	from
if	in	intersect	local	minus
mod	not	od	option	options
or	proc	quit	read	save
stop	then	to	union	while

Важной операцией в Maple является операция присваивания ($:=$). Синтаксис этого оператора следующий:

переменная := выражение;

Здесь в левой части задается имя переменной, а в правой части любое выражение, которое может быть числовым, символьным или просто другой переменной. Семантика (смысл) этого оператора в том, что переменной в левой части присваивается значение выражения, стоящего в правой части.

```

Объявление переменной:
> g;
                                     g
> whattype(g);
                                     string

```

Рис. 14 Переменная, которой не присвоено никакого значения (MF-14.GIF)

Переменные позволяют хранить и обрабатывать разнообразные типы данных, с которыми работает Maple. Мы уже знакомы с такими типами данных, как целый (*integer*), дробь (*fraction*), вещественный (*float*) и строка (*string*). Кроме этих типов данных существует еще большое множество типов, необходимых для выполнения аналитических преобразований: функция (*function*), индексные данные (*indexed*), множество (*set*), список (*list*), разложение (*series*), последовательность выражений (*exprseq*) и некоторые другие. Постепенно по мере продвижения нашего изучения Maple мы познакомимся со всеми этими типами данных.

По умолчанию переменная Maple имеет строковый тип, и ее значением является ее собственное имя. Поэтому простое объявление переменной g оператором $g;$ приведет к отображению в области вывода рабочего листа имени этой переменной, как показано на рис. 14.

На этом же рисунке можно увидеть функцию `whattype()`, которая определяет тип выражения или переменной, заданных в качестве ее параметра. Видно, что эта переменная действительно является строковой (`string`).

То, что переменная по умолчанию является строковой, оказывается очень полезным при задании функций. Если имя функции Maple задано не совсем правильно, или такой функции не существует, или не подключен пакет, где она расположена, то ответом Maple на попытку вычислить ее будет отображение в области вывода не результата выполнения функции, а полностью повторенная строка области ввода.

При присвоении переменной какого-нибудь значения, ее тип изменяется на тип присвоенного ей значения.

Переменные можно использовать для составления выражений наряду с числами. Все, сказанное выше о числовых выражениях и порядке их вычисления, относится и к выражениям, содержащим переменные.

Обычно в математических выражениях используются разнообразные математические функции. В Maple изначально определен большой набор стандартных математических функций, начиная от элементарных и заканчивая специальными функциями, используемыми при решении сложных задач математической физики. Справку о всех, имеющихся в Maple функций, можно получить, выполнив команду `?inifunction`.

В следующей таблице представлены основные математические функции и соответствующий им синтаксис Maple:

Функция	Синтаксис Maple
e^x	<code>exp(x)</code>
$\ln(x)$	<code>ln(x)</code> или <code>log(x)</code>
$\log_{10}(x)$	<code>log10(x)</code>
$\log_a(x)$	<code>log[a](x)</code>
\sqrt{x}	<code>sqrt(x)</code>
$ x $	<code>abs(x)</code>
$\operatorname{sgn}(x)$	<code>signum(x)</code>
$n!$	<code>n!</code>

Задание тригонометрических и гиперболических функций представлено в следующей таблице. Обратим внимание на несоответствие записи некоторых функций в русскоязычной мате-

матической литературе и в англоязычной, например функция тангенса угла. Значения параметров тригонометрических функций задаются в радианах.

Функция	Синтаксис Maple
$\sin(x)$	$\sin(x)$
$\cos(x)$	$\cos(x)$
$\operatorname{tg}(x)$	$\tan(x)$
$\operatorname{sec}(x)$	$\sec(x)$
$\operatorname{cosec}(x)$	$\csc(x)$
$\operatorname{ctg}(x)$	$\cot(x)$
$\operatorname{sh}(x)$	$\sinh(x)$
$\operatorname{ch}(x)$	$\cosh(x)$
$\operatorname{th}(x)$	$\tanh(x)$
$\operatorname{sech}(x)$	$\operatorname{sech}(x)$
$\operatorname{cosech}(x)$	$\operatorname{csch}(x)$
$\operatorname{cth}(x)$	$\operatorname{coth}(x)$

Задание обратных тригонометрических и обратных гиперболических функций представлено следующей таблицей:

Функция	Синтаксис Maple
$\operatorname{arcsin}(x)$	$\operatorname{arcsin}(x)$
$\operatorname{arccos}(x)$	$\operatorname{arccos}(x)$
$\operatorname{arctg}(x)$	arctan
$\operatorname{arcsec}(x)$	$\operatorname{arcsec}(x)$
$\operatorname{arccosec}(x)$	$\operatorname{arccsc}(x)$
$\operatorname{arcctg}(x)$	$\operatorname{arccot}(x)$
$\operatorname{arcsh}(x)$	$\operatorname{arcsinh}(x)$
$\operatorname{arcch}(x)$	$\operatorname{arccosh}(x)$
$\operatorname{arth}(x)$	$\operatorname{arctanh}(x)$
$\operatorname{arcsech}(x)$	$\operatorname{arcsech}(x)$
$\operatorname{arccosech}(x)$	$\operatorname{arccsch}(x)$
$\operatorname{arccth}(x)$	$\operatorname{arccoth}(x)$

Задание в Maple функций Бесселя, эллиптических интегралов, дельта-функции Дирака, функции Хевисайда и других специальных функций можно найти в Справке.

Команды

Технология работы в Maple заключается в том, что пользователь создает переменные и выражения и производит над ними некоторые действия в соответствии с алгоритмом решения поставленной задачи, используя стандартные функции или написанные собственные процедуры.

Синтаксис вызова стандартной команды или функции следующий:

```
команда(пар_1, пар_2, ... ,пар_n);
```

или

```
команда(пар_1, пар_2, ... ,пар_n):
```

Здесь команда - это имя вызываемой функции, а пар_1, пар_2, ... означают необходимые для выполнения команды параметры, которые могут быть переменными или даже выражениями, причем их тип должен соответствовать типу параметров соответствующей функции. Отметим, что первая форма задания команды (с завершающей точкой с запятой) осуществляет отображение результатов ее выполнения в области вывода, тогда как при второй форме (с завершающим двоеточием) команда выполняется, но никакого вывода результатов не происходит.

Система Maple является достаточно интуитивной системой, поэтому обыкновенно имя функции соответствует действию, которое она выполняет (следует учесть, что все имена заданы на английском языке). Например, ясно, что функция с именем `simplify()` осуществляет некоторые упрощения над выражением, заданным в качестве ее параметра.

Команды и функции, являющиеся частью ядра системы Maple, всегда доступны пользователю, тогда как для вызова других команд и функций необходимо подключить библиотеку или пакет, в которых они расположены. Для этого используются команды `readlib()` и `with()`. Первая подключает библиотеку, вторая - пакет. Параметром этих команд является имя библиотеки или пакета, функции которых пользователь желает использовать.

Команды и функции ядра

Наиболее часто используемые при аналитических преобразованиях команды и функции Maple располагаются в его системном ядре - части программного обеспечения системы аналитических вычислений, постоянно находящейся в памяти компьютера. К таким командам относятся команды, выполняющие разнообразные преобразования выражений, получающие решение уравнений и систем уравнений, дифференцирующие функции и т.д. В данном разделе вводятся команды, наиболее часто используемые при выполнении аналитических вычислений.

Команда `simplify()`

Начнем с команды упрощения выражений - команды `simplify()`. Эта команды предназначена для упрощения разнообразных выражений, включающих рациональные дроби (алгебраические выражения), содержащих тригонометрические, обратные тригонометрические

функции, логарифмы и экспоненты, т.е. с ее помощью можно попытаться упростить выражение, составленное из элементарных функций. Почему попытаться? Просто потому, что Maple может его упростить, а может и не упростить, так как он использует свои внутренние алгоритмы упрощения, которые могут содержать не все возможные формулы, используемые человеком при упрощении выражений, хотя практически он всегда упрощает заданное выражение.

Синтаксис команды упрощения следующий:

```
simplify(выражение);
```

В скобках в качестве параметра передается выражение, подлежащее упрощению. Команда `simplify()` ищет в выражении вызовы функций, квадратные корни, радикалы и степени и инициализирует подходящие процедуры упрощения. Maple содержит большое число подобных процедур, относящихся к упрощению выражений, содержащих определенный набор элементарных и специальных математических функций. Мы перечислим часть из них, остальные можно найти в справке по этой команде (например, установив курсор в рабочем листе на ее имя и нажав <F1>): `exp` - экспоненты, `ln` - логарифмы, `sqrt` - квадратные корни, `trig` - тригонометрические функции, `radical` - радикалы (дробные степени), `power` - степени, экспоненты, логарифмы, `BesselI` - функции Бесселя и т.д.

В вызове команды можно задать конкретные процедуры упрощения, и тогда только они будут использоваться для упрощения заданного выражения, а не весь возможный, установленный по умолчанию набор. Такой вызов обеспечивается следующим синтаксисом команды:

```
simplify(выражение, n1, n2, ...);
```

Здесь `n1`, `n2` и т.д. являются именами процедур упрощения: `Ei`, `ГАММА`, `RootOf`, `atsign`, `hypergeom`, `ln`, `polar`, `power`, `radical`, `siderel`, `sqrt`, `trig`. Полную информацию о формулах упрощения при использовании перечисленных значений параметров можно получить с помощью команды `?simplify[имя]`, где `имя` – одно из значений параметров функции упрощения.

При упрощении выражения можно предположить, что все переменные в нем являются, например, положительными, или принадлежат некоторому отрезку действительных чисел. Это осуществляется заданием ключевого параметра `assume=свойство`. Форма вызова команды в этом случае имеет вид:

```
simplify(выражение, assume=свойство);
```

где значение ключевого параметра свойство может принимать одно из следующих значений: `complex` - комплексная область, `real` - действительная область, `positive` - положительные действительные числа, `integer` - целые числа, `RealRange(a,b)` - интервал (a, b) действительных чисел.

Ниже представлены примеры использования команды упрощения выражений `simplify()`:

```
> f:= ln(exp(x));
```

$$f := \ln(e^x)$$

```
> simplify(f);
```

$$\ln(e^x)$$

```
> simplify(f, assume=real);
```

$$x$$

```
> a:=1/sqrt(5)*(((1+sqrt(5))/2)^3-((1-sqrt(5))/2)^3);
```

$$a := \frac{1}{5} \sqrt{5} \left(\left(\frac{1 + \sqrt{5}}{2} \right)^3 - \left(\frac{1 - \sqrt{5}}{2} \right)^3 \right)$$

```
> simplify(a);
```

$$2$$

Обратим внимание на упрощение выражения f . Использование команды без параметров не упростило выражения $\ln(e^x)$, тогда как второй оператор с предположением о действительной области изменения переменной x упростил заданное выражение. При упрощении Maple предполагает, что там, где это возможно, переменные изменяются в области комплексных чисел, тогда действительно невозможно упростить выражение f .

Команда `expand()`

Основное назначение команды `expand()` – представить произведение в виде суммы, т.е. данная команда раскрывает скобки в алгебраическом выражении. Она выполняется для любого полинома. Для частного двух полиномов (рациональная алгебраическая дробь) она раскрывает скобки в числителе и делит каждый член полученного выражения на знаменатель, с которым она не производит никаких преобразований.

Кроме того, данная команда умеет работать с большинством математических функций и знает, как раскрывать скобки в выражениях, содержащих следующие функции: $\sin(x)$, $\cos(x)$, $\operatorname{tg}(x)$, $\operatorname{sh}(x)$, $\operatorname{ch}(x)$, $\operatorname{th}(x)$, $\ln(x)$, специальные математические функции и т.д.

Синтаксис этой команды следующий:

```
expand(выр, выр1, выр2, ..., вырn);
```

где *выр* является выражением, в котором необходимо раскрыть скобки, необязательные параметры *выр1*, *выр2*, ..., *вырn* указывают системе, что в данных выражениях в заданном преобразуемом выражении *выр* раскрывать скобки не надо.

Несколько примеров использования этой команды приведены ниже:

```
> expand((x+1)*(x+2));
```

$$x^2 + 3x + 2$$

```
> expand((x+1)^3/(x+2)^2);
```

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

```
> expand(sin(x+y));
```

$$\sin(x)\cos(y) + \cos(x)\sin(y)$$

```
> expand(exp(a+ln(b)));
```

$$e^a b$$

```
> expand((x+1)^2*(y+z), x+1);
```

$$(x+1)^2 y + (x+1)^2 z$$

Как видно из приведенных примеров, данная функция может преобразовывать разнообразные тригонометрические выражения, выражения с экспоненциальными функциями и полиномы.

Команда `normal()`

Назначение команды `normal()` – упростить алгебраическую дробь, сократив и числитель, и знаменатель на наибольший общий делитель. Она имеет две формы вызова:

```
normal(f);
```


`normal(f, expanded);`

где f – алгебраическая дробь, а параметр `expanded` служит для указания того, что после сокращения дроби в числителе и знаменателе раскрываются скобки.

Если параметр f задан в виде списка, множества, последовательности, ряда, уравнения, отношения или функции, то команда `normal()` последовательно применяется к компонентам f . Например, в случае ряда, это означает, что упрощаются коэффициенты ряда.

Приведем несколько примеров использования команды `normal()`:

`> f:=(x^2-y^2)/(x-y)^3;`

$$f := \frac{x^2 - y^2}{(x - y)^3}$$

`> normal(f);`

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

`> normal(f, expanded);`

$$\frac{x + y}{x^2 - 2xy + y^2}$$

`> normal(2/x + y/3 = 0);`

$$\frac{1}{3} \frac{6 + xy}{x} = 0$$

`> s:=sin(x*(x+1)-x);`

$$s := \sin(x(x + 1) - x)$$

`> normal(s);`

$$\sin(x^2)$$

Обратим внимание на то, что третья команда в этих примерах применяется к уравнению (equation) – типу Maple, обсуждение которого отложим до знакомства с командой решения уравнений и неравенств.

Вычисление выражений и переменных сложных типов

Maple по умолчанию стремится производить все преобразования и вычисления с максимальной точностью, т.е. там, где это возможно, использовать арифметику с дробными числами.

Иногда это не совсем удобно, и необходимо получить результат в форме привычных десятичных чисел. Для вычисления и преобразования коэффициентов выражений, представленных обыкновенными дробями, существует команда `evalf()`, аппроксимирующая дробные числа десятичными числами, а также вычисляющая выражение и константы Maple в форме числа с плавающей точкой.

По умолчанию используется арифметика чисел с плавающей точкой с 10 значащими цифрами. Это значение можно изменить, переустановив системную константу `Digits` в другое значение, например,

```
> Digits:=15;
```

После выполнения этого оператора все последующие в сеансе Maple вычисления с плавающей точкой будут производиться с 15 значащими цифрами. Обратим внимание, что имя этой системной переменной начинается с прописной буквы.

Синтаксис команды `evalf()` следующий:

```
evalf(f);
```

```
evalf(f, n);
```

где `f` – вычисляемое выражение, а `n` – число значащих цифр, используемых при вычислении.

Несколько примеров этой команды приводится ниже:

```
> Pi;
```

$$\pi$$

```
> evalf(Pi, 20);
```

$$3.1415926535897932385$$

```
> Digits:=30;
```

$$Digits:= 30$$

```
> evalf(5/3*exp(-2)*sin(Pi/4));
```

$$.159494160850684873267979942007$$

```
> evalf(cos(1));
```

$$.540302305868139717400936607443$$

```
> evalf(3/4*x^2+1/3*x-sqrt(2), 10);
```

$$.7500000000 x^2 + .3333333333 x - 1.414213562$$

Maple вычисляет значение переменной простого типа равным тому алгебраическому выражению, которое ей присвоено, но значения переменных сложных, или составных типов (массивы, таблицы, векторы), вычисляются в Maple равным имени этой переменной. Для того, чтобы выяснить действительное значение, содержащееся в такой переменной, в Maple используется команда `eval()`.

Например, переменная, содержащая массив, является переменной сложного типа, и для отображения ее содержимого следует использовать команду `eval()`. В следующем примере иллюстрируется задание массива и отображение его содержимого:

```
> a:=array(1..2,1..2,[ [1,2],[Pi,Pi]]);
```

$$a:=\begin{bmatrix} 1 & 2 \\ \pi & \pi \end{bmatrix}$$

```
> a;
```

a

```
> eval(a);
```

$$\begin{bmatrix} 1 & 2 \\ \pi & \pi \end{bmatrix}$$

Как видно из приведенного примера, простое задание в области ввода имени переменной, содержащей массив, приводит к отображению в области вывода имени самой переменной.

Данную команду следует использовать с переменными любого сложного типа для выяснения их содержимого.

Команда `subs()`

При выполнении математических преобразований часто необходимо произвести замену переменных в выражении, функции, уравнении и т.д., то есть вместо какой-то переменной подставить ее представление через некоторые другие переменные. Для этих целей в системе Maple существует команда `subs()`, синтаксис которой следующий:

```
subs( старое_выражение = новое_выражение , выражение );
```

```
subs( s1, ... , sn, выражение );
```

где каждое из `s1, ... , sn` является уравнением или множеством/списком уравнений. (Уравнение – два выражения, соединенные знаком равенства (=).)

Первая форма команды анализирует выражение, выделяет в нем все вхождения старое_выражение и подставляет вместо них новое_выражение. Вторая форма позволяет выполнить серию подстановок в выражение. Подстановки выполняются последовательно, начиная с $s1$. Это означает, что после выполнения первой подстановки, определенной уравнением $s1$, Maple отыскивает вхождения левой части уравнения $s2$ во вновь полученном выражении и заменяет каждое такое вхождение на выражение, заданное в левой части уравнения $s2$. Если подстановки заданы в виде списка или множества уравнений, то они выполняются одновременно, т.е. вхождения выражений, заданных в левых частях уравнений, определяются в исходном параметре выражение. Например, после выполнения подстановки `subs(x=y, y=x, [x,y])`, исходный список $[x, y]$ будет преобразован в $[x, x]$, тогда как при использовании команды `subs({x=y, y=x}, [x,y])` переменные x и y в списке поменяются местами: список будет иметь вид $[y, x]$.

Здесь следует немного сказать о внутреннем представлении объектов Maple. Каждый объект Maple, в том числе и выражение, делится на подобъекты первого уровня, которые в свою очередь, также делятся на подобъекты и т.д. Этот процесс продолжается до тех пор, пока не дойдем до базисных элементов Maple (объектов основных типов: целые, вещественные, дроби и т.д.). В результате получается ветвящаяся, древоподобная структура представления любого объекта Maple через базовые элементы. Для выражения $x^7 + 8x^6 + \sqrt{x^2} - 9$ древовидная структура представлена на рис. 15. Каждый подобъект соответствующего уровня отображен в рамке. Здесь также можно увидеть операции, которые выполняются между подобъектами одного уровня.

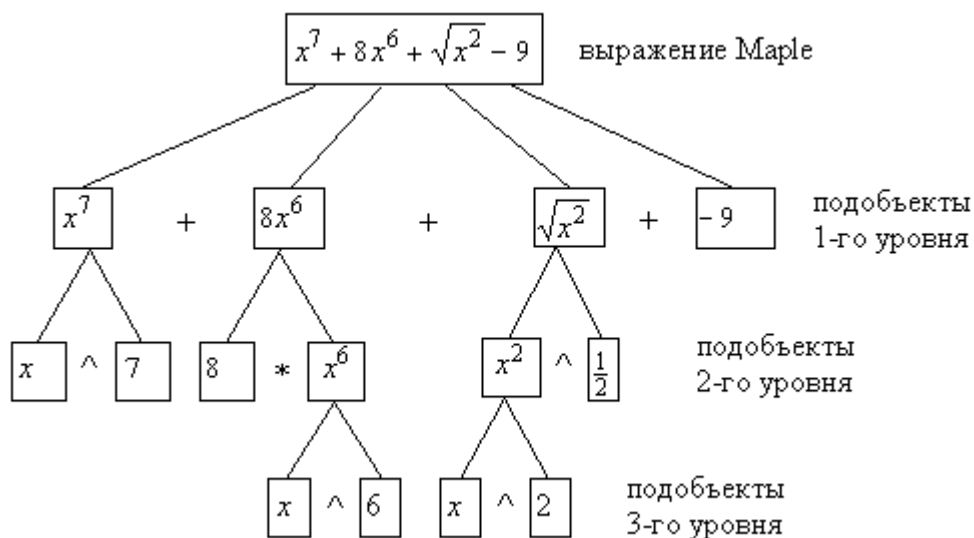


Рис. 15 Представление выражения Maple в виде структуры (MF-15.GIF)

Команда `pops(объект)` определяет количество объектов первого уровня, а команда `op(объект)` выдает их в виде последовательности выражений. Эта же команда позволяет извлечь подобъекты первого уровня, указав в качестве первого параметра порядковый номер подобъекта первого уровня. Ниже представлены результаты применения этих команд к выражению $x^7 + 8x^6 + \sqrt{x^2} - 9$:

```
> p := x^7+8*x^6+sqrt(x^2)-9;
```

$$p := x^7 + 8x^6 + \sqrt{x^2} - 9$$

```
> pops(p);
```

4

```
> s:=op(p);
```

$$s := x^7, 8x^6, \sqrt{x^2}, -9$$

```
> op(3,p);
```

$$\sqrt{x^2}$$

Теперь можно вернуться к описанию команды подстановки `subs()`. Так вот, эта команда заменяет выражение, заданное в левой части одного из первых параметров, на выражение из правой части, только если это выражение из левой части совпадает с одним из подобъектов в структурном представлении выражения, в котором производится подстановка. Такая подстановка называется « синтаксической подстановкой ». Поэтому, например, подстановка в выражение s^3 вместо s^2 выражения $1 - c^2$ с помощью команды `subs(s^2=1-c^2, s^3)` не приведет к желаемому результату.

Однако не стоит паниковать, так как в Maple имеется команда `algsubs(старое_выражение = новое_выражение, выражение)`, предназначенная для алгебраической подстановки и которая выполнит не решаемую командой `subs` вышеуказанную подстановку. Подробную информацию об этой команде можно, как всегда, получить из Справки Maple.

Команда `solve()`

Команда `solve()` одна из самых полезных команд системы аналитических вычислений Maple. Практически решение ни одной задачи не обходится без решения какого-нибудь уравнения или системы уравнений, равенства или системы неравенств. Именно это и вы-

полняет вышеназванная команда. Но прежде чем перейти к синтаксису этой команды, следует еще немного остановиться на типах данных, используемых в Maple.

Мы знакомы с оператором присваивания $:=$, который позволяет присвоить переменной в левой части числовое значение или выражение, стоящее в правой части.

Два выражения, соединенные знаком равенства $=$, представляют новый тип данных Maple - *уравнение* (equation). Уравнение можно присвоить обычной переменной Maple.

```
> 2*x^2+5=x+x^4;
```

$$2x^2 + 5 = x + x^4$$

```
> whattype("");
```

=

```
> f:=2*x^2+5=x+x^4;
```

$$f := 2x^2 + 5 = x + x^4$$

```
> whattype(f);
```

=

При проверке типа переменной, значением которой является уравнение, с помощью команды `whattype()` результатом является равенство $=$, означающее, что тип проверяемой переменной является уравнением.

Точно также, как и при задании уравнений, два выражения, соединенные знаками \geq (больше или равно), \leq (меньше или равно), $>$ (больше) или $<$ (меньше), представляют новый тип - *неравенство* (inequation).

В Maple для хранения некоторого набора выражений и/или переменных определенного типа используется конструкция специального типа - *множество* (set). Для его задания достаточно заключить в фигурные скобки последовательность выражений и/или переменных:

```
{m, s=x^2+s^2};
```

Если переменной присвоено значение, являющееся множеством, то для получения значения элемента множества следует использовать индекс - заданный сразу же после имени переменной в квадратных скобках номер элемента множества. Например, чтобы получить значение второго элемента множества, определенного выше, следует использовать следующую последовательность команд Maple:

```

> M:={m, s=x^2+s^2};

                               M:= {m,s = x^2 + s^2}

> M[2];

                               s = x^2 + s^2

> whattype(M[2]);

                               =

> whattype(M);

                               set

```

Последние два оператора показывают тип переменной M и тип значения ее второго элемента, так как эта переменная является множеством.

Во многих командах Maple в качестве параметра или в качестве получаемого значения используется еще один тип данных – *список* (list), который представляет собой последовательность значений или выражений, заключенную в квадратные скобки:

```
[m, s=x^2+s^2];
```

Этот тип данных используется для хранения коэффициентов полиномов, коэффициентов разложения функций в разнообразные ряды и т.п.

Теперь можно перейти к описанию синтаксиса команды `solve()`, позволяющей решать уравнения и системы уравнений, неравенства и системы неравенств. Ее синтаксис, как и синтаксис всех команд Maple, достаточно прост и легко запоминается:

```
solve(уравнение, переменная);
```

```
solve({уравнение1, уравнение2, ...}, {переменная1, переменная2, ...});
```

Первая форма команды предназначена для решения одного уравнения относительно заданной переменной, тогда как вторая форма позволяет решать системы уравнений относительно заданных во втором параметре переменных. Обратим внимание на то, что система уравнений и ее неизвестные переменные задаются в виде множеств. Результатом в этом случае является также множество значений неизвестных в виде уравнений, тогда как в случае задания одного уравнения результатом будет выражение (в случае одного корня уравнения) или последовательность выражений (в случае нескольких корней). Если не задана переменная/переменные, относительно которых следует решать уравнение/систему уравнений, то Maple выдаст все

решения относительно всех неопределенных переменных в исходных уравнениях. Следующий пример иллюстрирует некоторые из приведенных ситуаций:

```
> eq:=x^2-2*x+y^2=0;
```

$$eq:=x^2-2x+y^2=0$$

```
> solve(eq,x);
```

$$1+\sqrt{1-y^2}, 1-\sqrt{1-y^2}$$

```
> solve({eq},x);
```

$$\{x=1+\sqrt{1-y^2}\}, \{x=1-\sqrt{1-y^2}\}$$

```
> eq1:=x+y=0;
```

$$eq1:=x+y=0$$

```
> solve({eq,eq1},{x,y});
```

$$\{x=0, y=0\}, \{y=-1, x=1\}$$

Когда Maple не может найти решение, или решение не существует, то в результате выполнения команды `solve()` возвращается пустая последовательность со значением `NULL`, и в случае, если Maple не может найти решение (хотя оно существует), глобальная переменная `_SolutionsMayBeLost` устанавливается равной `true`.

В общем случае полиномиальное уравнение степени выше 4 может не иметь решения, выраженного с помощью радикалов. В этом случае для представления результатов Maple использует специальную функцию `RootOf(уравнение, переменная)`, которая применяется для обозначения любого корня уравнения относительно заданной переменной:

```
> eq:=x^4+x^3+1=0;
```

$$eq:=x^4+x^3+1=0$$

```
> solve(eq,x);
```

$$\text{RootOf}(_Z^4+_Z^3+1)$$

```
> evalf(s);
```

$$-1.018912794 - .6025654200 I$$

В этом примере функция $\text{RootOf}(_Z^4 + _Z^3 + 1)$ представляет любое решение уравнения $_Z^4 + _Z^3 + 1 = 0$. Обратим внимание на переменную $_Z$. Это системная переменная, сгенерированная Maple для представления целых чисел. Подобные переменные используются Maple для представления всех решений тригонометрических уравнений. Кроме указанной переменной также используются переменная $_NN$ для неотрицательных целых значений и $_B$ для двоичных значений (0 или 1). Обратим внимание, что с помощью функции $\text{evalf}()$ можно получить приближенные числовые значения функции RootOf .

Если решение уравнения выражается достаточно сложной формулой, то Maple может ввести специальные «промежуточные» переменные вида $\%n$, где n является целым числом, для обозначения каких-либо выражений и использовать эти переменные в окончательном представлении решения. Подобная ситуация иллюстрируется следующим примером:

```
> solve(x^3+x+1=0,x);
```

$$-\frac{1}{6}\%2 + \frac{2}{(108 + 12\sqrt{93})^{1/3}}, \frac{1}{12}\%2 - \frac{1}{(108 + 12\sqrt{93})^{1/3}} + \frac{1}{2}I\sqrt{3}\left(-\frac{1}{6}\%2 - \frac{2}{(108 + 12\sqrt{93})^{1/3}}\right),$$

$$\frac{1}{12}\%2 - \frac{1}{(108 + 12\sqrt{93})^{1/3}} - \frac{1}{2}I\sqrt{3}\left(-\frac{1}{6}\%2 - \frac{2}{(108 + 12\sqrt{93})^{1/3}}\right)$$

$$\%1 = \frac{1}{(108 + 12\sqrt{93})^{1/3}}$$

$$\%2 = (108 + 12\sqrt{93})^{1/3}$$

Здесь Maple вводит две переменные $\%1$ и $\%2$, причем первую в окончательном выражении решения кубического уравнения он не использует. Эти переменные можно в дальнейшем использовать как обычные переменные с присвоенными значениями как если бы пользователь задал их сам.

Для получения всех решений тригонометрических уравнений следует задать значение глобальной переменной $_EnvAllSolutions$ равным true . По умолчанию Maple решает тригонометрическое уравнение на промежутке $[-\pi, \pi]$. Следующий пример иллюстрирует использование глобальной переменной $_EnvAllSolutions$.

```
> eq:=sin(x)^2+2*sin(x)+1=0;
```

$$eq:=\sin(x)^2+2\sin(x)+1=0$$

> s:=solve(eq,x);

$$s:=-\frac{1}{2}\pi$$

> _EnvAllSolutions:=true;

_EnvAllSolutions:= true

> s:=solve(eq,x);

$$s:=-\frac{1}{2}\pi+2\pi_Z1 \sim$$

Как видно, в случае `_EnvAllSolutions:=true` Maple действительно выдает все решения тригонометрического уравнения с использованием целочисленной системной переменной `_Z1~`, в которой знак тильда (\sim) означает, что на значения переменной наложены некоторые ограничения. В данном случае эта переменная может принимать только целочисленные значения.

При решении систем уравнений, когда ответ получается в виде множества уравнений, в которых левая часть является неизвестной переменной, для того, чтобы присвоить найденные значения переменным, относительно которых решалась система, следует применять команду `assign()`. Эта команда присваивает переменным, стоящим в левой части уравнений из множества решений, значения, равные правым частям.

> eq:=x*a+y*b=c;

$$eq:= xa + yb = c$$

> s:=solve({eq,x+y=1},{x,y});

$$s:=\left\{y=\frac{-a+c}{-a+b},x=\frac{c-b}{-a+b}\right\}$$

> assign(s); x; y;

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

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

После того, как система решена и получено решение в виде множества равенств, значения искомых переменных не определены, поэтому применяется команда `assign()` для присваивания этим переменным полученных решений.

Если решения получено в виде последовательности выражений, то получить значение соответствующего решения можно с помощью индекса.

```
> eq:=x^4+2*x^2+1=0;
```

$$eq: = x^4 + 2x^2 + 1 = 0$$

```
> s:=solve(eq);
```

$$s: = I, -I, I, -I$$

```
> x1:=s[2]; x1;
```

$$x1: = -I$$

$$-I$$

Напомним, что в приведенном примере I означает комплексную мнимую единицу, равную $\sqrt{-1}$.

Команда `fsolve()`

По умолчанию Maple пытается найти аналитическое выражение для корней уравнения. Если это ему не удастся, то, как отмечалось выше, он просто ничего не печатает в области вывода. В подобных случаях (если корни действительно существуют) можно воспользоваться командой `fsolve()`, которая находит численное решение уравнения или системы уравнений. Формат команды отличается от формата команды `solve()` наличием третьего параметра опция:

```
fsolve(уравнения, переменные, опция);
```

где задание первых двух параметров соответствует заданию аналогичных параметров в команде `solve()`, а параметр опция может принимать следующие значения:

Значение	Смысл
<code>complex</code>	разыскиваются комплексные корни (только для полиномов)
<code>fulldigits</code>	используется арифметика с максимальной мантисой

<code>maxsols=n</code>	разыскивается n решений (только для полиномов)
<code>a..b</code> или <code>x=a..b</code>	задан промежуток $[a,b]$, на котором разыскивается решение (во второй форме задания этой опции <code>x</code> обозначает имя неизвестной переменной в уравнении)

По умолчанию для произвольного уравнения эта функция находит одно решение, но для полиномов определяются все действительные корни. Чтобы найти все корни полинома, включая комплексные, следует задать опцию `complex`. Несколько примеров использования команды численного решения уравнений показано ниже:

```
> eq:=x^4+2*x^2-1=0;
                                eq:= x4 + 2x2 - 1
> s:=fsolve(eq,x);
                                s:= -.6435942529, .6435942529
> s:=fsolve(eq,x,complex);
                                s:= -.6435942529, -1.553773974I, 1.553773974I, .6435942529
> fsolve(ln(sin(x))=0,x);
                                1.570796327
> fsolve(ln(sin(x))=0,x,x=2..infinity);
                                7.853981634
> fsolve(ln(sin(x))=0,x,x=8..infinity);
                                14.13716694
```

В этом примере также показано каким образом можно последовательно находить корни произвольного уравнения, задавая интервал изменения неизвестной величины с учетом полученного решения на предыдущем шаге нахождения корня (последние три команды).

Дифференцирование и интегрирование

Maple позволяет вычислять обыкновенные и частные производные аналитического выражения по одной или нескольким переменным. Для этой процедуры предназначены команды

`diff()` и `Diff()`. Вторая команда является так называемой *отложенной* командой (`inert command`), которая не вычисляет производную от выражения, а просто отображает математическую запись взятия производной. Результат действия отложенной команды можно присвоить переменной Maple, а в дальнейшем при помощи команды `value()` вычислить результат этой отложенной команды. Отложенная форма команды дифференцирования удобна, когда необходимо видеть, какие операции были сделаны для получения необходимого выражения. Кроме отложенной команды дифференцирования в Maple есть и еще ряд отложенных форм некоторых команд, полную информацию о которых можно получить в Справке.

Синтаксис команды дифференцирования следующий:

```
diff(выражение, переменная_1, переменная_2, ... , переменная_n);
```

```
diff(выражение, [переменная_1, переменная_2, ... , переменная_n]);
```

В результате выполнения любой из приведенных команд будет вычислена частная производная n -го порядка от заданного первым параметром выражения по заданным n переменным.

При формировании производных высокого порядка полезен оператор последовательности `$`, который позволяет проще и нагляднее задать производную. Например, для вычисления третьей производной функции $f(x)$ по переменной x можно использовать команду `diff(f(x), x, x, x)`, в которой три раза указано дифференцирование по переменной x , или применить в команде дифференцирования оператор последовательности `x$3`, что упрощает и делает более наглядным задание третьей производной: `diff(f(x), x$3)`. Несколько примеров использования команды дифференцирования приведены ниже:

```
> f:=x^2*sin(x)+sqrt(y)*ln(cos(x));
```

$$f := x^2 \sin(x) + \sqrt{y} \ln(\cos(x))$$

```
> diff(f,x);
```

$$2x \sin(x) + x^2 \cos(x) - \frac{\sqrt{y} \sin(x)}{\cos(x)}$$

```
> diff(f,x$2);
```

$$2 \sin(x) + 4x \cos(x) - x^2 \sin(x) - \sqrt{y} - \frac{\sqrt{y} \sin(x)^2}{\cos(x)^2}$$

```
> diff(f,x,y);
```

$$-\frac{1}{2} \frac{\sin(x)}{\sqrt{y} \cos(x)}$$

```
> fDerive:=Diff(f,x);
```

$$fDerive:=\frac{\partial}{\partial x}(x^2 \sin(x) + \sqrt{y} \ln(\cos(x)))$$

```
> g:=sqrt(fDerive);
```

$$g:=\sqrt{\frac{\partial}{\partial x}(x^2 \sin(x) + \sqrt{y} \ln(\cos(x)))}$$

```
> value("");
```

$$\sqrt{2x \sin(x) + x^2 \cos(x) - \frac{\sqrt{y} \sin(x)}{\cos(x)}}$$

Последние три команды показывают использование отложенной формы команды дифференцирования.

Интегрирование выражений по заданной переменной осуществляется командой `int()`, которая также имеет отложенную форму `Int()`. Эта команда позволяет вычислять как неопределенный интеграл от выражения (при этом в ответе не будет никакой постоянной интегрирования) с использованием следующего синтаксиса команды:

```
int( выражение , переменная );
```

так и определенный интеграл с помощью следующего синтаксиса команды

```
int( выражение , переменная=a..b );
```

где a и b являются пределами интегрирования, причем эти пределы могут быть и аналитическими выражениями. Несколько примеров использования команды интегрирования приведены ниже:

```
> f:=a*x^2*sin(b*x);
```

$$f:=ax^2 \sin(bx)$$

```
> int(f,x);
```

$$\frac{(-b^2 x^2 \cos(bx) + 2 \cos(bx) + 2bx \sin(bx))a}{b^3}$$

```
> int(f,x=0..1);
```

$-\frac{a(b^2 \cos(bx) - 2 \cos(b) - 2b \sin(b))}{b^3} - 2 \frac{a}{b^3}$ <p>> int(f,x=0..a);</p> $-\frac{a(b^2 a^2 \cos(ab) - 2 \cos(ab) - 2ab \sin(ab))}{b^3} - 2 \frac{a}{b^3}$ <p>> Int(f,x=0..Pi);</p> $\int_0^{\pi} a x^2 \sin(bx) dx$ <p>> value("");</p> $-\frac{a(b^2 \pi^2 \cos(\pi b) - 2 \cos(\pi b) - 2\pi b \sin(\pi b))}{b^3} - 2 \frac{a}{b^3}$
--

Для символьного вычисления определенного интеграла существуют две опции, управляющие обработкой разрывов подынтегральной функции. Эти опции задаются третьим параметром в командах `int()` или `Int()`.

По умолчанию команда интегрирования проверяет выражение на непрерывность в области интегрирования и вычисляет интеграл как сумму отдельных определенных интегралов на промежутках непрерывности функции. Опция `continuous` отключает этот режим и вычисляет интеграл как разность значений первообразной подынтегральной функции в точке начала и конца промежутка интегрирования.

Еще одна опция `CauchyPrincipalValue` вычисляет несобственные интегралы первого и второго рода в смысле главного значения Коши.

Если Maple не может найти замкнутую форму выражения для определенного интеграла, то команда интегрирования возвращает просто вызов самой себя (в области вывода печатается математическая запись вычисления интеграла, как при обращении к отложенной команде интегрирования). В подобных случаях можно вычислить значение определенного интеграла численным способом с помощью команды `evalf()`. Синтаксис подобной конструкции следующий:

```
evalf( int(f,x=a..b) );
```

```
evalf( Int(f,x=a..b) );
```

```
evalf( Int(f,x=a..b), digits, flag );
```

Параметр `digits` позволяет задать число значащих цифр при вычислениях приближенного значения интеграла (по умолчанию это число равно числу значащих цифр, определенных значением системной константы `Digits`).

По умолчанию используется квадратурная формула Кленшо-Куртиса (Clenshaw-Curtis). Если в подынтегральном выражении встречается сингулярность, то применяется специальная методика символьного анализа для разрешения существующей сингулярности. Для задач с неустранимыми сингулярностями используется адаптивный метод двойных экспоненциальных квадратур. Параметр `flag` позволяет явно задать метод численного интегрирования. Он может принимать следующие значения:

Параметр <code>flag</code>	Смысл
<code>_Csqquad</code>	применяется только квадратура Кленшо-Куртиса без вызова процедуры обработки сингулярности
<code>_Dexp</code>	применяется адаптивный метод двойных экспоненциальных квадратур
<code>_Ncrule</code>	применяется квадратурная формула Ньютона-Котеса, являющийся методом фиксированного порядка, и не эффективен для высоких точностей (<code>Digits > 15</code>)

Несколько примеров численного вычисления интегралов помогут освоиться с использованием вышеприведенной методики:

```
> int(sin(x)*ln(x),x=0..1);
```

$$\text{Ci}(1) - \gamma$$

```
> evalf( int(sin(x)*ln(x),x=0..1) );
```

$$-.2398117420$$

```
> Int(sin(x)*ln(x),x=0..1)=evalf( Int(sin(x)*ln(x),x=0..1, 20, _Dexp ) );
```

$$\int_0^1 \sin(x) \ln(x) dx = -.23981174200056319789$$

```
> Int(1/(1+x^2),x=0..infinity)= evalf( Int(1/(1+x^2),x=0..infinity, 30, _Dexp ) );
```


$$\int_0^{\infty} \frac{1}{1+x^2} dx = 1.57079632679489661923132169164$$

> Int(exp(x-x^2/2)/(1+exp(x)/2),x=-infinity..infinity)=

evalf(Int(exp(x-x^2/2)/(1+exp(x)/2),x=-infinity..infinity));

$$\int_{-\infty}^{\infty} \frac{e^{-\frac{1}{2}x^2}}{1+\frac{1}{2}e^x} dx = 1.805577049$$

В заключение этого раздела отметим, что в системе Maple имеется набор команд для полного исследования функций: `limit()` – для отыскания предела функции, `sum()` – для нахождения всевозможных конечных сумм, `series()` – для разложения функций в ряды Тейлора, Маклорена и Лорана, `extrema()` – для исследования экстремумов функций как одной, так многих переменных, `minimize()` и `maximize()` – для поиска минимума и максимума функции на заданном промежутке. Описание всех этих и других команд можно, естественно, найти в Справке Maple.

Пакеты

Пакеты в Maple используются для удобства организации работы пользователя. Пакет представляет собой набор команд для решения задач, относящихся к определенным разделам математики, или решения определенных задач графического представления информации, например, пакет `finance` служит для решения задач финансовой математики, в пакете `stats` собраны команды для статистической обработки результатов и т.д.

Для того, чтобы использовать команды какого-нибудь пакета, необходимо подключить его, так как они все находятся не в ядре системы Maple, а в специальных файлах. Подключение пакетов осуществляется с помощью команды

`with(пакет);`

где в качестве параметра указывается имя соответствующего пакета. Может оказаться, что подключаемый пакет содержит команду с таким же именем, что и в ранее подключенном пакете. В этом случае в области вывода отображается сообщение о переопределении соответствующей команды, и результат ее действия будет соответствовать команде, находящейся в последнем подключенном пакете.

Полный список и описание пакетов можно найти в справочной системе Maple, или выполнить команду

```
?index[package];
```

С пакетом графического вывода результатов (`plots`) мы познакомимся в разделе *Графика*, а пакеты финансовой математики (`finance`) и статистической обработки результатов (`stats`) будут описаны во второй части книги.

Графика

Системы аналитических вычислений привлекают исследователей не только своими возможностями реализации алгоритмов построения аналитических решений, но и развитой графикой, начиная от построения простейших двумерных кривых и заканчивая сложными трехмерными поверхностями и анимацией двумерных и трехмерных изображений. В любой момент пользователь может отобразить результаты своих исследований в виде графических образов, которые, как известно, более информативны, чем скупые ряды цифр. (Хотя в некоторых ситуациях цифра может оказаться более полезной, чем общая картина изменения какого-либо параметра.)

Исходными данными для графических команд, объединенных в пакете `plots`, могут выступать как типы данных, поддерживаемые Maple (функции, массивы, множества, графические структуры), так и данные других программ и приложений, сохраненные в файлах текстового формата.

Чтобы можно было использовать команды графического отображения этого пакета, необходимо подключить его с помощью оператора `with(plots)`.

Простейшая графическая команда `plot()` расположена в системной библиотеке Maple, и поэтому доступна в любое время. Именно с нее мы и начнем знакомство с графическими возможностями системы аналитических вычислений Maple.

Команды двумерной графики

Наиболее употребительной командой двумерной графики является многофункциональная команда `plot()`. Она позволяет рисовать график одной или нескольких функций одной переменной, заданных в явном или параметрическом виде, а также отображать множество точек в декартовой или полярной системе координат. Синтаксис команды `plot()` следующий

```
plot(f, h, v, опции);
```

где f – функция, которую необходимо отобразить, h , v – соответственно диапазон изменения независимой переменной по горизонтальной оси графика и диапазон изменения значения функции по вертикальной оси графика. Отметим, что диапазон изменения независимой переменной задается в виде $x=a..b$, где a и b наименьшее и наибольшее значение изменения переменной, а x имя независимой переменной. Если диапазон не задан, т.е. второй параметр представляет собой просто имя независимой переменной в функции, то по умолчанию принимается следующий интервал ее изменения $-10..10$. Этот параметр обязательно должен присутствовать при задании графика командой `plot()`. Вертикальный диапазон ограничивает вывод графика заданной областью изменения функции. Этот параметр необязателен, как и опции, задающиеся в виде уравнений `имя_опции=значение`. При отсутствии явного задания опций принимаются значения опций по умолчанию.

Опции задают вид отображаемого графика: толщину, цвет и тип линии графика, вид осей координат, размещение надписей и т.д. Набор возможных опций во всех командах двумерного графического вывода, за некоторым исключением, одинаков. Ниже перечислены все опции и соответствующие им значения (умалчиваемые значения подчеркнуты):

- **title**=`строка` – задает заголовок рисунка. По умолчанию заголовок не выводится.
- **coords**=<значение> – по умолчанию при выводе как явно заданной функции, так и параметрически заданной функции используется декартова система координат. Задание значения данной опции меняет тип системы координат. Возможные значения следующие: `bipolar`, `cardiod`, `cassinian`, `elliptic`, `hyperbolic`, `invcassinian`, `invelliptic`, `logarithmic`, `logcosh`, `maxwell`, `parabolic`, `polar`, `rose` и `tangent`, описание которых можно получить в справочной системе Maple с помощью команды `?coords`. Здесь отметим только, что значение `polar` задает полярную систему координат.
- **axes**=<значение> – определяет тип отображаемых осей координат. Эта опция может принимать следующие значения: `normal` - обычные оси координат; `boxed` - график заключен в прямоугольник с нанесенными шкалами по нижней и вертикальной левой граням; `frame` - оси с точкой пересечения в левом нижнем углу рисунка; `none` - оси не отображаются.
- **scaling**=<значение> – масштаб, в котором отображается график. Если значение этой опции равно `constrained`, то это соответствует заданию абсолютных значений по осям координат, т.е. одна единица измерения по оси независимой переменной равна одной еди-

нице измерения по оси значений функции. Умалчиваемое значение равно unconstrained, и это соответствует тому, что оси растягиваются таким образом, чтобы их размеры соответствовали размерам графического окна вывода.

- **style**=<значение> – задание отображения графика функции линиями (значение опции равно line) или точками (значение опции равно point). Значения параметра `patch` и `patchnograd` применяются только тогда, когда выводится замкнутый многоугольник. В этом случае его внутренняя область закрашивается цветом, установленным в опции `color`. Если в графическом выводе нет замкнутых многоугольников, то действие этих значений опции соответствуют значению line.
- **numpoints**=*n* – число вычисляемых точек, по которым строится график (значение по умолчанию равно 50)
- **resolution**=*n* – горизонтальное разрешение дисплея в пикселях (значение по умолчанию равно 200)

- **color**=<значение> – задание цвета, которым отображается график. В качестве значения этой опции может выступать одно из зарезервированных значений цвета в Maple: `aquamarine`, `black`, `blue`, `navy`, `ral`, `cyan`, `brown`, `gold`, `green`, `gray`, `grey`, `khaki`, `magenta`, `maroon`, `orange`, `pink`, `plum`, `red`, `sienna`, `tan`, `turquoise`, `violet`, `wheat`, `white` и `yellow`. Можно также определить и свой собственный цвет, соответствующий смещению заданных частей красного, зеленого и синего цветов. Это осуществляется с помощью следующей команды

```
macro(palegreen=COLOR(RGB, .5607, .7372, .5607)),
```

где `palegreen` имя константы нового цвета, в котором красный составляет 0.5607 части, зеленый - 0.7372 и синий - 0.5607.

- **font**=<значение> – задание шрифта для вывода текста на рисунке. Значение опции задается в виде списка [`семейство`, `стиль`, `размер`]. Параметр `семейство` задает гарнитуру шрифта: `TIMES`, `COURIER`, `HELVETICA` или `SYMBOL`. Параметр `стиль` определяет стиль шрифта: для гарнитуры `TIMES` возможные значения `ROMAN`, `BOLD`, `ITALIC` или `BOLDITALIC`, для гарнитур `COURIER` и `HELVETICA` стиль можно опустить, или задать `BOLD`, `OBLIQUE` или `BOLDOBLIQUE`, для шрифта `SYMBOL` стиль не задается. Последний параметр `размер` задает размер шрифта в пунктах (один пункт приблизительно равен 1/72 дюйма).

- **axesfont**=<значение> – задание шрифта для надписей под засечками вдоль осей координат. Значение этой опции аналогично значению опции `font`.
- **labelfont**=<значение> – задание шрифта для названия осей координат. Значение этой опции аналогично значению опции `font`.
- **titlefont**=<значение> – задание шрифта для заголовка рисунка. Значение этой опции аналогично значению опции `font`.
- **linestyle**=*n* – задание типа линии графика. Значением этой опции является целое число *n*. При *n*=0 или 1 график отображается сплошной линией. Если значение больше 2, то используются разнообразные шаблоны линий для вывода графика функции: штриховые, с разной толщиной и т.д.
- **thickness**=*n* – задание толщины линии графика, меняется от 0 до 15, соответствуя изменению толщины линии от самой тонкой до самой жирной.
- **symbol**=<значение> – задание типа символа, которым помечаются точки графика функции при `style=point`. Может принимать следующие значения: `box` для ‘`⊙`’, `cross` для ‘`+`’, `circle` для ‘`⋅`’, `POINT` (прописными буквами) для ‘`·`’ (точка) и `diamond` для ‘`◊`’.
- **labels**=[*x*,*y*] – задание названий осей координат. Параметры *x* и *y* задаются в виде строк и соответствуют отображаемым названиям горизонтальной и вертикальной осей. По умолчанию принимают значения имени независимой переменной и имени функции.
- **xtickmarks**=*n* – задает число точек, не менее которых должно быть помечено на горизонтальной оси. Значение этой опции может быть целым числом или списком значений координат точек горизонтальной оси.
- **ytickmarks**=*n* – задает число точек, не менее которых должно быть помечено на вертикальной оси. Значение этой опции может быть целым числом или списком значений координат точек вертикальной оси.

Большинство этих опций можно изменить с помощью команд меню интерфейса пользователя или команд контекстного меню, отображаемого нажатием левой кнопки мыши в области рисунка. Как отмечалось в первой главе при выделении в тексте документа графики или получении фокуса окна с графическим выводом, если задан режим вывода графики в отдельном окне, меняется основное меню в строке меню интерфейса, а также заменяется контекстная панель инструментов таким образом, чтобы обеспечить доступ к командам интерфейса, работающим с полученным графическим отображением.

На рис. 15 показан общий вид интерфейса пользователя с меню и контекстной панелью инструментов для работы с графикой. Также на этом же рисунке отображено контекстное меню, появляющееся при щелчке правой кнопкой мыши, когда курсор находится в области графического вывода.

Меню **Insert** (Вставка), **Format** (Формат) и **Options** (Опции) заменяются новыми меню **Style** (Стиль), **Axes** (Оси), **Projection** (Проекция) и **Animation** (Анимация), которые позволяют изменить основные опции для вывода графики. Все команды этих меню дублируются в контекстном меню, а часть в контекстной панели инструментов для двумерной графики.

Команды **Line** (Линия), **Point** (Точка), **Patch** (Заливка) и **Patch w/o grid** (Заливка без сетки) меню **Style** устанавливают значение опции `style` равной соответственно `line`, `point`, `patch` или `patchnograd`. На контекстной панели инструментов этим командам соответствуют первые четыре кнопки (см. рис. 15).

Подменю **Symbol** (Символ), **Line Style** (Тип линии) и **Line Width** (Толщина линии) позволяют установить значения опций `symbol`, `linestyle` и `thickness`. Кнопок на контекстной панели инструментов для этих опций не предусмотрено.

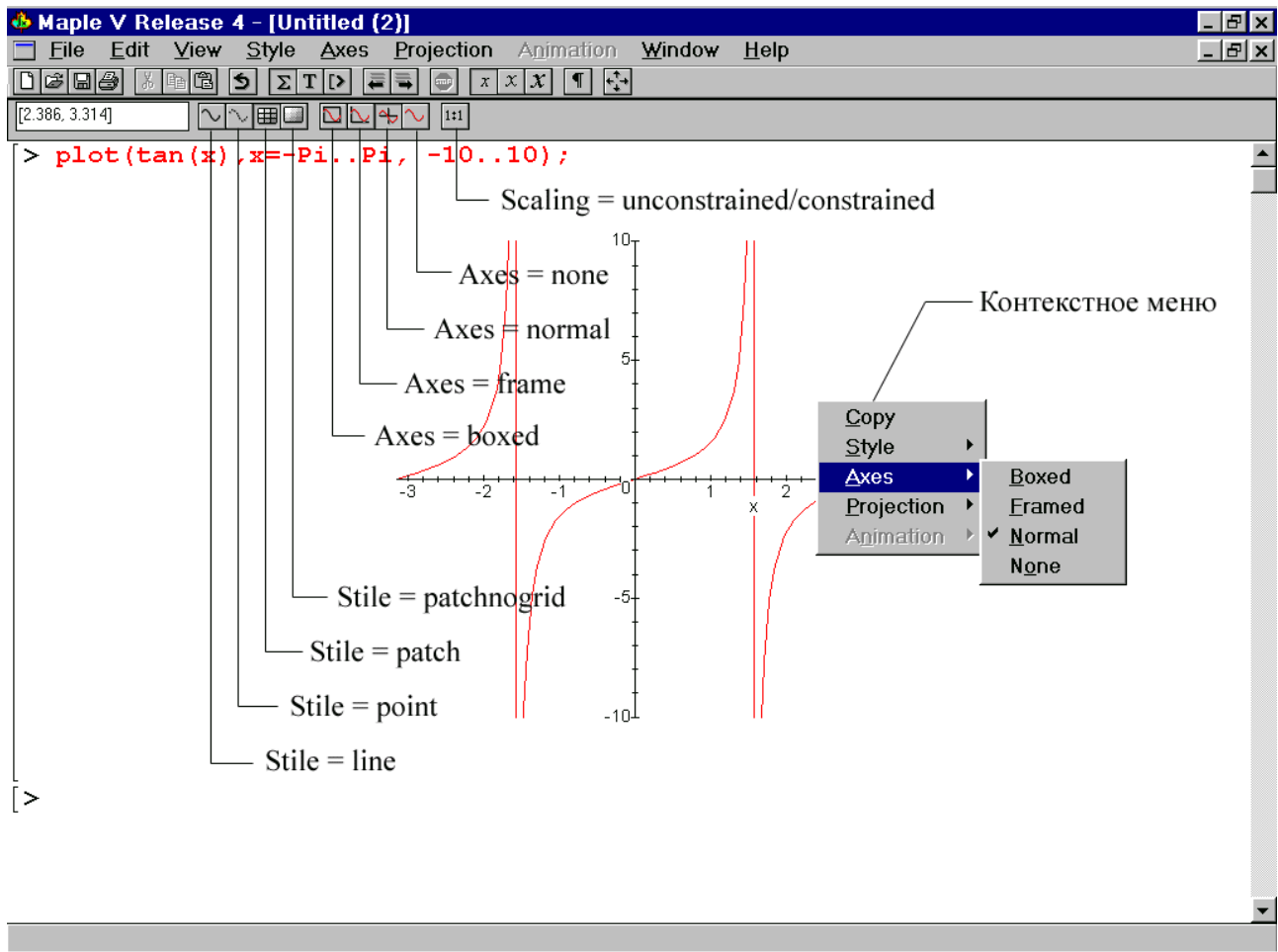


Рис. 16 Интерфейс Maple при работе с графикой (MF-16.GIF)

Команды меню **Axes** (Оси) соответствуют установке значений `boxed`, `frame`, `normal` и `none` опции `axes`. На контекстной панели инструментов им соответствуют следующие четыре кнопки (рис. 15).

Меню **Projection** (Проекция) устанавливает значения опции `scaling`. На панели инструментов значения этой опции можно устанавливать с помощью последней кнопки. Если она не нажата, то соответствует значению `unconstrained`, если нажата - установлено значение `constrained`.

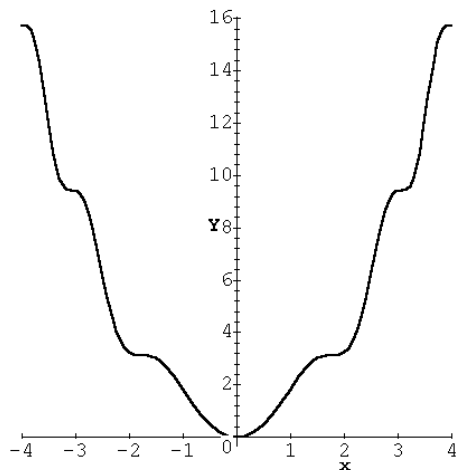
Последнее меню **Animation** специально предназначено для анимации изображений и становится доступным, когда в документе Maple графика выводится командой создания анимации `animate()`.

Все перечисленные команды меню можно выполнить из контекстного меню, в котором кроме этих команд присутствует команда **Copy** (Копировать), копирующая графический рисунок в Буфер обмена операционной системы Windows 95/NT для вставки его в документ другого приложения или обработки какой-либо графической программой. Отметим, что командой **Copy** меню **Edit** (Правка) нельзя поместить рисунок, расположенный на рабочем листе, в Буфер обмена, если только он не выделен с помощью стандартных операций Windows 95/NT (с помощью мыши или клавишами `<Shift>+<→>`). Если графика выводится в отдельном окне системы, эта команда доступна сразу же без выделения рисунка.

Несколько примеров помогут вам освоиться с этой простой в использовании командой отображения графики:

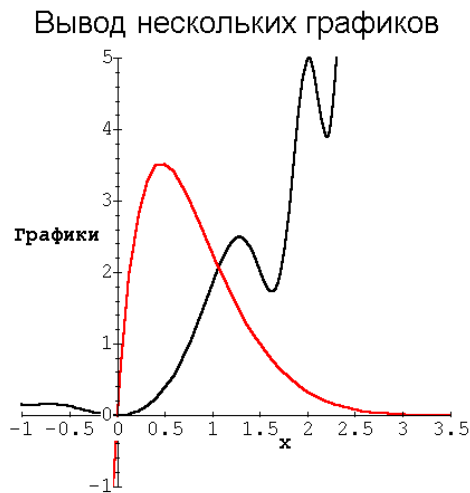
```
> plot(x^2+sin(x^2),x=-4..4,
>      color=black,
>      title=`Пример вывода графика`,
>      titlefont=[HELVETICA,16],
>      xtickmarks=8,
>      thickness=2,
>      axesfont=[COURIER,11],
>      labels=[`x`,`Y`],
>      labelfont=[COURIER,BOLD,11]);
```

Пример вывода графика



Для вывода нескольких функций на одном графике необходимо в команде `plot()` задавать функции в виде множества, а значение опции `color` в виде списка позволяет задать цвет для вывода графиков функций:

```
> plot({x^2+sin(x^3), 20*exp(-2*x)*sin(x)},
>      x=-1..3.5, -1..5,
>      color=[black,red],
>      title=`Вывод нескольких графиков`,
>      titlefont=[HELVETICA,16],
>      xtickmarks=8,
>      thickness=2,
>      axesfont=[COURIER,11],
>      labels=[`x`,`Графики`],
>      labelfont=[COURIER,BOLD,11]);
```

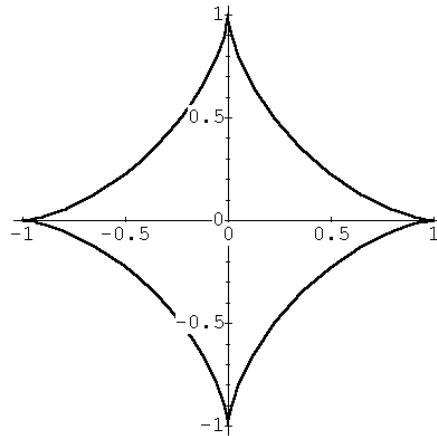



Две функции $y = x^2 + \sin(x^3)$ и $y = 20 e^{-2x} \sin(x)$ заданы в виде множества. Значение опции `color` определяет цвет графика первой функции черным, а второй – красный.

Вывод параметрически заданной кривой (или несколько таких функций на одном графике) ничем не отличается от вывода явно задаваемых функций. Единственное отличие заключается в том, что параметрическая кривая задается в виде списка, где первый и второй элементы являются значениями соответственно горизонтальной и вертикальной координат, а третий элемент списка задает изменение параметра в принятом в Maple виде. Следующий пример демонстрирует отображение параметрически заданной кривой:

```
> plot([cos(t)^3,sin(t)^3,t=0..2*Pi],
>      color=[black],
>      title=`Вывод параметрически заданной кривой`,
>      titlefont=[HELVETICA,12],
>      xtickmarks=4,
>      thickness=2,
>      axesfont=[COURIER,11]);
```

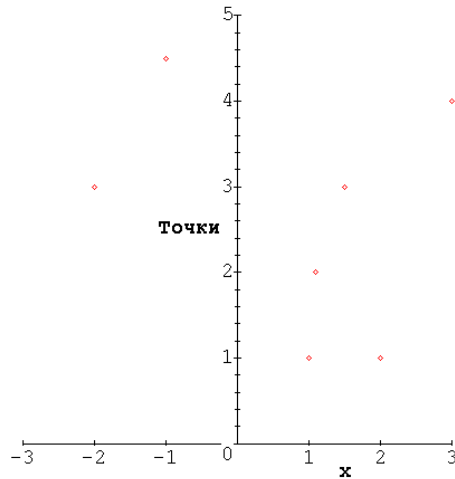
Вывод параметрически заданной кривой



Команда `plot()` позволяет отображать на графике отдельные точки, которые задаются в виде списка списков, т.е. списка, элементами которого являются списки. Эти элементы задают координаты точек по горизонтали и по вертикали. Для вывода точек необходимо задать значение опции `style` равной `POINT`. Если этого не сделать, то Maple отобразит ломаную линию, соединяющую точки в последовательности их задания. Следующий пример отображает точки на плоскости в виде символа ромб `symbol=diamond`:

```
> plot([[1,1],[2,1],[3,4],[-2,3],[-1,4.5],[1.5,3],[1.1,2]],
>       x=-3..3, 0..5,
>       color=[red],
>       style=POINT,
>       symbol=diamond,
>       title=`Вывод точек командой PLOT`,
>       titlefont=[HELVETICA,16],
>       xtickmarks=4,
>       thickness=2,
>       axesfont=[COURIER,11],
>       labels=[`x`,`Точки`],
>       labelfont=[COURIER,BOLD,11]);
```

Вывод точек командой PLOT



В пакете `plots` существует большой набор команд для отображения кривых на плоскости. Со всеми этими командами можно ознакомиться в справочной системе Maple. Здесь мы только кратко перечислим возможности двумерного графического вывода в системе Maple: построение графика функции в логарифмическом масштабе по вертикальной оси (`logplot()`), построение графика функции в логарифмическом масштабе по обеим осям координат (`log-logplot()`), построение замкнутого многоугольника, заданного своими вершинами (`polygonplot()`), изображение конформного отображения комплексной функции (`conformal()`), вывод текстовой строки (`textplot()`), изображение линии уровня функции двух переменных в прямоугольнике (`implicitplot()`), изображение решения дифференциального уравнения (`odeplot()`) и др. Все эти команды становятся доступными после подключения пакета `plots` командой `with(plots)`.

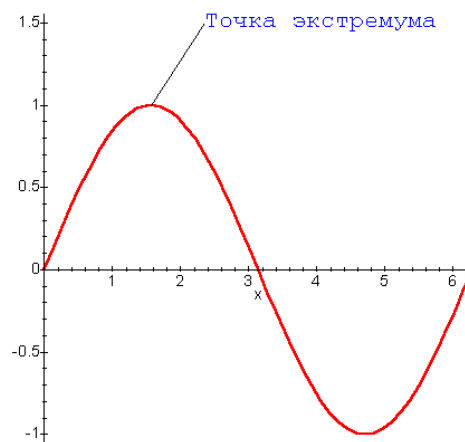
Часто необходимо на одном графике совместить результаты отображения, осуществляемого разными графическими командами. Для этого результат действия каждой команды должен быть присвоен некоторой переменной. Заметим, что при этом никакого графического вывода не происходит. Команда `display()`, находящаяся в пакете `plots`, выводит «отложенную» и сохраненную в переменных Maple графику. Синтаксис этой команды следующий:

```
display([pic1, pic2, ...], <опции>);
```

Здесь элементы списка являются либо явно заданными графическими образами с помощью команд вывода графики, либо переменные, в которых хранятся графические образы. Опции соответствуют рассмотренным ранее опциям команды `plot()`. Следует отметить опцию `insequence`, которая применима только для этой команды. Если ее значение равно `true`, то графические образы `pic1`, `pic2`, ... будут последовательно отображаться один за другим, составляя таким образом кадры анимации.

Следующий пример дает представление о том, как используется команда `display()`. В переменной `graph1` сохраняется график функции $\sin(x)$, переменная `graph2` содержит вывод выносной линии для надписи, а переменная `graph3` хранит надпись, выводимую рядом с выносной линией.

```
> with(plots):
> graph1:=plot(sin(x),x=0..2*Pi,thickness=2):
> graph2:=plot([[Pi/2,1],[Pi/2+Pi/4,1.5]],color=black):
> graph3:=textplot([[Pi/2+Pi/4,1.5,`Точка экстремума`]],
>                 color=blue, align=RIGHT,
>                 font=[COURIER,12]):
> display([graph1,graph2,graph3]);
```



Результирующий график, составленный из этих трех графических изображений, выводится с помощью команды `display()`, в которой в качестве элементов списка заданы вышеупомянутые переменные, хранящие графические образы.

Введение	2
Графический интерфейс пользователя	4
Общий вид	4
Рабочие листы	5
Область ввода	6
Область вывода	9
Вывод графики	9
Работа с меню	11
Виды меню.....	13
Основное меню.....	13
Справочная система.....	17
Организация справки.....	18
Вызов справки с помощью меню Help.....	19
Вызов справки из рабочего листа.....	20
Основы работы	22
Объекты, переменные и выражения.....	22
Команды.....	28
Команды и функции ядра.....	29
Команда <code>simplify()</code>	29
Команда <code>expand()</code>	31
Команда <code>normal()</code>	32
Вычисление выражений и переменных сложных типов.....	33
Команда <code>subs()</code>	35
Команда <code>solve()</code>	37
Команда <code>fsolve()</code>	43
Дифференцирование и интегрирование	44
Пакеты.....	49
Графика	50
Команды двумерной графики	50