МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Е.В. Лебедева

Использование интегрированных символьных систем при решении экономико-математических задач

Учебно-методическое пособие
(для студентов экономического профиля)

Орел -2012

УДК 339.138

ББК 65.290-2

Печатается по решению редакционно-издательского совета Орловского государственного университета (протокол № ….от……..2012)

Научный редактор: доктор педагогических наук, профессор В. Д. Селютин

Рецензенты: доктор педагогических наук, профессор Тарасова О.В.
кандидат экономических наук, доцент Бураева Е. В.

Лебедева Елена Валерьевна Использование интегрированных символьных систем при решении экономико-математических задач: Учебно-методическое пособие (для студентов экономического профиля). – Орел: ФГБОУ ВПО «ОГУ», 2012.– 49 с.

ISBN 978-5-9929-0044-6

Данная книга является учебно-методическим пособием по дисциплинам «*Интегрированные символьные системы*», «*Интегрированные математические пакеты на ЭВМ*», «*Символьные вычисления*». Пособие представляет собой практическое руководство по изучению возможностей пакета аналитических вычислений *Maple*. Подробные теоретические сведения чередуются с практическими заданиями из профессиональной области будущего экономиста. Последовательное изучение тем и выполнение заданий позволит шаг за шагом освоить основные приемы работы в математической системе *Maple*.

Учебное пособие предназначено для студентов физико-математических и экономических направлений и специальностей вузов, а также для аспирантов и научных работников, использующих математические методы и модели в научных исследованиях.

Оглавление

Введение

Данное пособие представляет собой практическое руководство по изучению возможностей пакета символьных вычислений *Maple* для решения задач в различных отраслях науки, финансов и экономики, математики и статистики. Следует отметить, что эта книга предназначена в первую очередь для обучения студентов решению экономико-математических и финансовых задач на персональном компьютере при помощи *Maple*. Задачи и упражнения, приведенные в качестве примеров, соответствуют программам по курсу общей математики, математической экономики, математических методов в экономике для студентов направлений прикладная информатика, прикладная математика и информатика, бизнес-информатика, экономика вузов.

Структура пособия: пособие состоит из шести тем. В каждой теме содержатся:

– теоретическая часть – в ней приведено описание изучаемых команд *Maple*;

– практические задания – подробное пошаговое описание действия команд *Maple* на конкретных примерах по математике, математической экономике и экономике; эти задания предназначены для выполнения студентами под руководством преподавателя;

– контрольные вопросы – предназначены для закрепления теоретического материала.

Глава I. Среда *Maple*

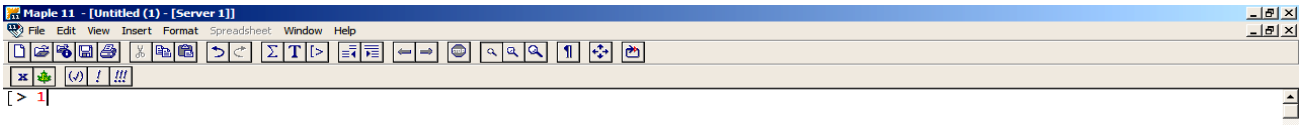
§1. Структура окна *Maple*

Maple − это пакет для аналитических вычислений на компьютере, содержащий более двух тысяч команд, которые позволяют решать задачи алгебры, геометрии, математического анализа, дифференциальных уравнений, статистики, математической физики.

Для того, чтобы запустить *Maple*, необходимо в *Главном меню Windows* выбрать в группе *Программы* название данного приложения: *Maple*.

Maple представляет собой типичное окно *Windows*, которое состоит из *Строки названия*, *Основного меню*, *Панели инструментов*, *Рабочего поля* и *Строки состояния*, а также *Линейки* и *Полос прокрутки*.

Вид фрагмента окна *Maple11 и 12*, содержащего *Строку названия*, *Основное меню*, *Панель инструментов*:



Пункты *Основного меню*:

File (Файл) − содержит стандартный набор команд для работы с файлами, например: сохранить файл, открыть файл, создать новый файл и т.д.

Edit (Правка) − содержит стандартный набор команд для редактирования текста, например: копирование, удаление выделенного текста в буфер обмена, отмена команды и т.д.

View (Вид) – содержит стандартный набор команд, управляющих структурой окна *Maple*.

Insert (Вставка) – служит для вставки полей разных типов: математических текстовых строк, графических двух и трехмерных изображений.

Format (Формат) – содержит команды оформления документа, например: установка типа, размера и стиля шрифта.

Options (Параметры) – служит для установки различных параметров ввода и вывода информации на экран, принтер, например, таких как качество печати.

Windows (Окно) – служит для перехода из одного рабочего листа в другой.

Help (Справка) – содержит подробную справочную информацию о *Maple*.

Работа в *Maple* проходит в режиме сессии – пользователь вводит предложения (команды, выражения, процедуры), которые воспринимаются условно и обрабатываются *Maple*. Рабочее поле разделяется на три части:

1) область ввода - состоит из командных строк. Каждая командная строка начинается с символа >;

2) область вывода - содержит результаты обработки введенных команд в виде аналитических выражений, графических объектов или сообщений об ошибке;

3) область текстовых комментариев - содержит любую текстовую информацию, которая может пояснить выполняемые процедуры.

Текстовые строки не воспринимаются *Maple* и никак не обрабатываются.

Для того, чтобы переключить командную строку в текстовую,

следует на *Панели инструментов* нажать мышью на кнопку Обратное переключение текстовой строки в командную осуществляется нажатием на *Панели инструментов* на кнопку

.

§2. Арифметические операции. Целые и рациональные числа, константы в *Maple*

Математические константы и арифметические операции.

Основные математические константы:

Pi – число π ; **I** – мнимая единица *i*; **infinity** – бесконечность; **true**, **false** – логические константы, обозначающие истинность и ложность высказывания.

Знаки арифметических операций:

+ - сложение; **–** - вычитание;

***** - умножение; / - деление;

^ - возведение в степень; **!** – факториал.

Знаки сравнения: **<, >, >=,<=, <>, =.**

Комплексные, целые и рациональные числа.

Числа в *Maple* бывают действительные (real) и комплексные (complex). Комплексное число записывается в алгебраической форме $z=x+iy$, и в командной строке такая запись должна выглядеть так:

> z:=x+I*y;

Вещественные числа разделяются на целые и рациональные. Целые числа (integer) выражаются цифрами в десятичной записи. Рациональные числа могут быть представлены в 3-х видах:

1) рациональной дроби с использованием оператора деления, например: **28/70**;

2) с плавающей запятой (float), например: **2.3;**

3) в показательной форме, например: **1.602*10^(-19)** означает $1{,}602 \cdot 10^{-19}$

Для того, чтобы получить рациональное число не в точной форме, а в виде приближенного значения (числа с плавающей запятой), следует дописывать к целой части числа .0. Пример:

> **75/4;**

$$\frac{75}{4}$$

> 75/4.0;

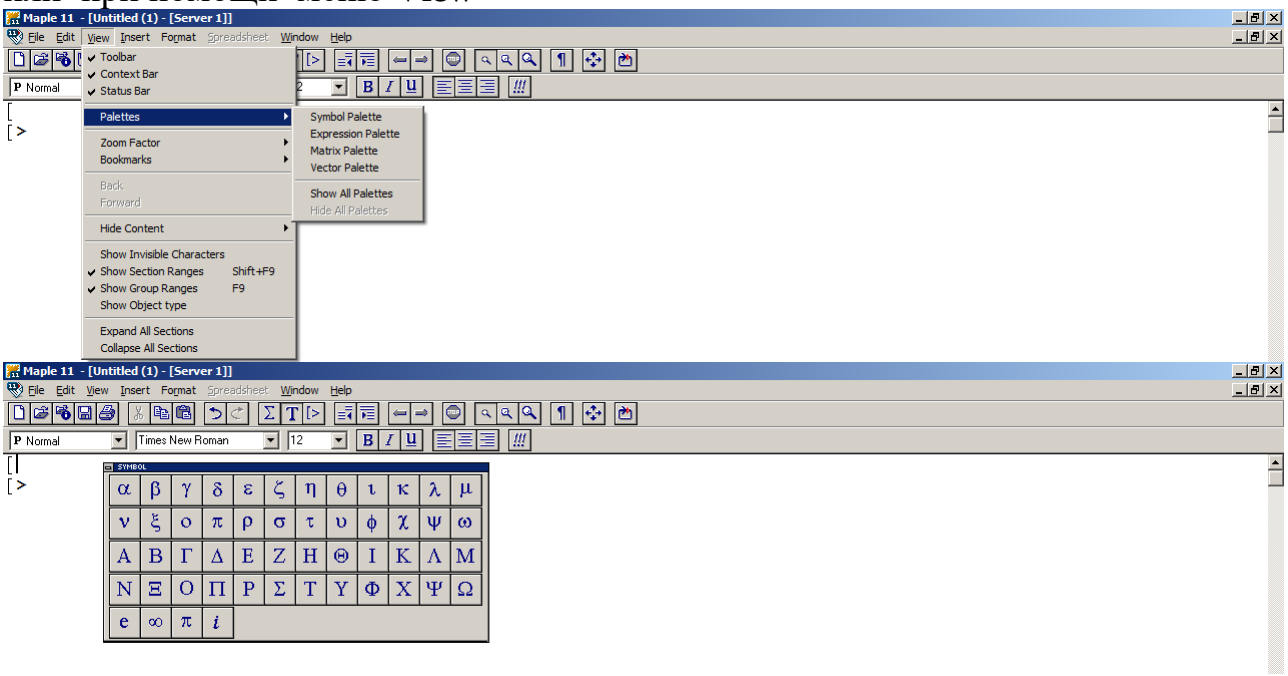
18.75000000

В *Maple* можно записать буквы греческого алфавита в полиграфическом виде. Для этого в командной строке набирается название греческой буквы. Например, буква α получится, если набрать **alpha**.

Таблица строчных греческих букв и их названий:

α – alpha	β – beta	γ – gamma	δ – delta
ε – epsilon	ς – zeta	η – eta	θ - theta
ι – ita	κ – kappa	λ – lambda	ν - nu
μ – mu	ξ –xi	π – pi	ρ - rho
σ – sigma	υ – upsilon	φ – phi	χ - chi
ψ – psi	ω -omega		

Заглавные греческие буквы можно записать, если набирать название греческой буквы с заглавной, например, чтобы получить Ω, следует набрать **Omega**. Греческие буквы также можно набирать с помощью специального меню или при помощи меню View



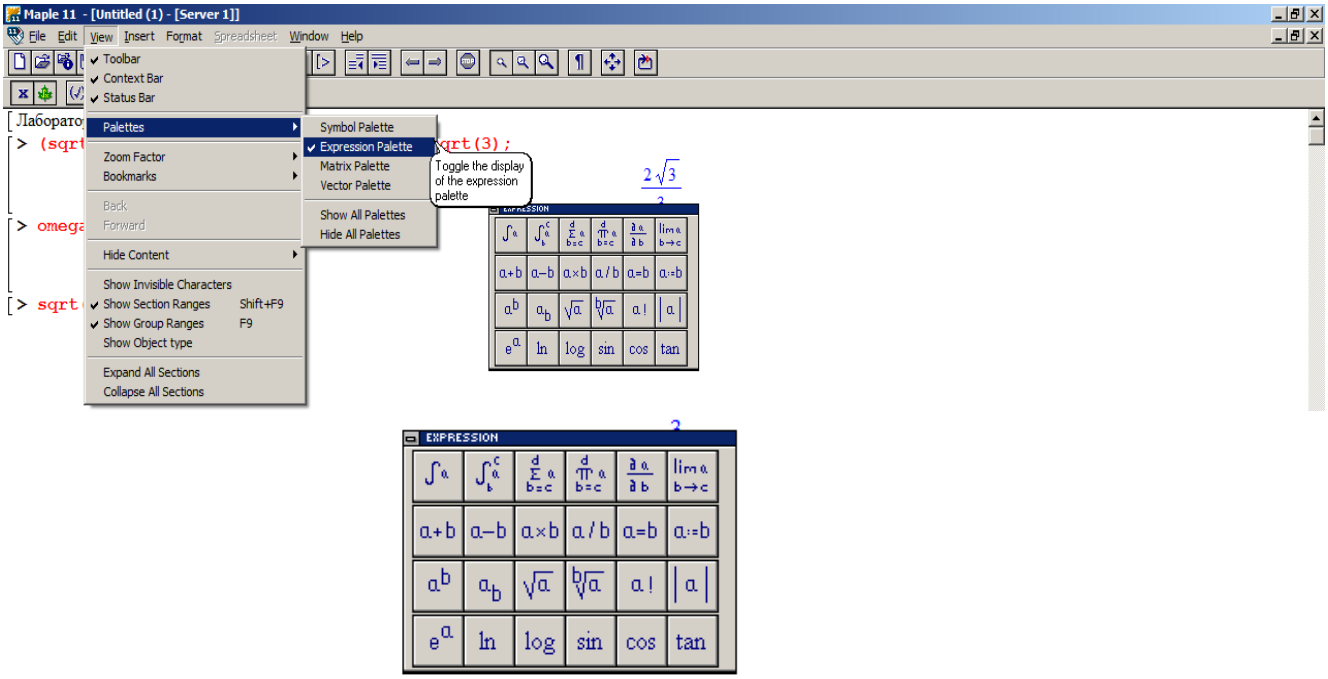
Пример. Вычислить значение $\dfrac{\sqrt{6+2\sqrt{5}}-\sqrt{6-2\sqrt{5}}}{\sqrt{3}}$.

Для этого в командной строке набираем:

> (sqrt(6+2*sqrt(5))-sqrt(6-2*sqrt(5)))/sqrt(3); и нажимаем *Enter*. В результате получится точное значение $\frac{2}{3}\sqrt{3}$ 3

Пример. Набрать формулы $\omega = \dfrac{\theta}{t}$ и $|f(x)-\delta| < \varepsilon$. Для этого в командной строке набираем:

> **omega=theta/t; abs(f(x)-delta)<epsilon;** нажимаем *Enter* или при помощи меню View



§3. Синтаксис команд. Стандартные функции

Синтаксис команд.

Стандартная команда *Maple* состоит из имени команды и ее параметров, указанных в круглых скобках: **command(p1, p2, …)**. В конце каждой команды должен быть знак (**;**) или (**:**). Разделитель (**;**) означает, что в области вывода после выполнения этой команды будет сразу виден результат. Разделитель (**:**) используется для отмены вывода, то есть когда команда выполняется, но ее результат на экран не выводится.

Символ процента (**%**) служит для вызова предыдущей команды. Этот символ играет роль краткосрочной замены предыдущей команды с целью сокращения записи. Пример использования (**%**):

> **a+b;**

$$a+b$$

> **%+c;**

$$a+b+c.$$

Для присвоения переменной заданного значения используется знак присвоить (**:=**).

Когда программа *Maple* запускается, она не имеет ни одной команды, полностью загруженной в память. Большая часть команд имеют указатели их нахождения, и при вызове они загружаются автоматически. Другие команды находятся в стандартной библиотеке и перед выполнением обязательно должны быть вызваны командой **readlib(command)**, где **command** – имя вызываемой команды.

Остальная часть процедур *Maple* содержится в специальных библиотеках подпрограмм, называемых пакетами. Пакеты необходимо подгружать при каждом запуске файла с командами из этих библиотек.

Имеется два способа вызова команды из пакета:

1) можно загрузить весь пакет командой **with(package)** где

package – имя пакета;

2) вызов какой-нибудь одной команды **command** из любого пакета **package** можно осуществить, если набрать команду в специальном формате:

> **package[command](options);**

где вначале записывается название пакета **package**, из которого надо вызвать команду, а затем в квадратных скобках набирается имя самой команды **command**, и после чего в круглых скобках следуют параметры **options** данной команды.

К библиотекам подпрограмм *Maple* относятся, например, следующие пакеты: **linalg** – содержит операции линейной алгебры; **geometry** – решение задач планиметрии; **geom3d** – решение задач стереометрии; **student** – содержит команды, позволяющие провести поэтапное решение задачи в аналитическом виде с промежуточными вычислениями.

Стандартные функции.

Математическая запись	Запись в *Maple*
e^x	**exp(x)**
$\ln x$	**ln(x)**
$\lg x$	**log10(x)**
$\log_a x$	**log[a](x)**
$\sqrt{x}$	**sqrt(x)**
$\lvert x \rvert$	**abs(x)**
$\sin x$	**sin(x)**
$\cos x$	**cos(x)**
tgx	**tan(x)**
$ctgx$	**cot(x)**
$\sec x$	**sec(x)**
$\operatorname{cosec} x$	**csc(x)**
$\arcsin x$	**arcsin(x)**
$arc\cos x$	**arccos(x)**
$arctgx$	**arctan(x)**
$arcctgx$	**arccot(x)**
shx	**sinh(x)**
chx	**cosh(x)**

tgx	**tanh(x)**
cthx	**coth(x)**

Maple содержит огромное количество специальных функций, таких, как Бесселевы функции, Эйлеровы бета- и гамма – функции, интеграл ошибок, эллиптические интегралы, различные ортогональные полиномы.

С помощью функции **exp(x)** определяется число *e*=2.718281828… посредством записи **exp(1)**.

Пример. Вычислить $\cos\frac{\pi}{3}+tg\frac{14\pi}{3}$. Для этого набираем в командной строке:

> cos(Pi/3)+tan(14*Pi/3);

Нажимаем *Enter*. В результате в области вывода появиться число: $-\frac{2}{3}\sqrt{3}$.

Пример. Вычислить $\sin^4\frac{\pi}{8}+\cos^4\frac{3\pi}{8}+\sin^4\frac{5\pi}{8}+\cos^4\frac{7\pi}{8}$.

Для этого наберите в командной строке:

> combine((sin(Pi/8))^4+(cos(3*Pi/8))^4+ (sin(5*Pi/8))^4+ (cos(7*Pi/8))^4);

Нажимаем *Enter*. (значение команды **combine** – преобразовывать выражения, например, со степенями). В результате в области вывода должно появиться число: $\frac{3}{2}$.

§4. Преобразование экономических и математических выражений

Maple обладает широкими возможностями для проведения аналитических преобразований математических формул. К ним относятся такие операции, как приведение подобных, разложение на множители, раскрытие скобок, приведение рациональной дроби к нормальному виду и многие другие.

Выделение частей выражений.

Математическая формула, над которой будут производиться преобразования, записывается в следующей форме: **> eq:=exp1=exp2;** где **eq** – произвольное имя выражения, **exp1** – условное обозначение левой части формулы, **exp2** – условное обозначение правой части формулы.

Выделение правой части выражения осуществляется командой **rhs(eq)**, выделение левой части выражения – командой **lhs(eq)**.

Рассмотрим *пример:* прибыль *P* определяется как разница между доходом *R* и затратами *C*: $P = R - C = 480Q_1 - Q_1^2 + 340Q_2 - 1{,}5Q_2^2 - 50000$

> P:=R-C=480*Q1-Q1^2+340*Q2-1.5*Q2^2-50000;

$$P := R - C = 480\ Q1 - Q1^2 + 340\ Q2 - 1.5\ Q2^2 - 50000$$

> lhs(P);

$$R - C$$

> rhs(P);

$$480\ Q1 - Q1^2 + 340\ Q2 - 1.5\ Q2^2 - 50000\ .$$

Если задана рациональная дробь вида *a/b*, то можно выделить ее числитель и знаменатель с помощью команд **numer** и **denom**, соответственно. *Пример:* фирма выпускает товар, спрос на который выражается функцией $y = \frac{200}{x+2}$, где *x* – цена товара

> y:=200/(x+2);

$$y := \frac{200}{x+2}$$

> numer(y);

$$200$$

> denom(y);

$$x + 2\ .$$

Тождественные преобразования выражений.

Раскрытие скобок выражения **eq** осуществляется командой **expand(eq)**. *Пример:* функция спроса на товар имеет вид $d = (1-4p)(25+8p^2)$

>d:=(1-4*p)*(25+8*p^2);

$$d = (1 - 4\ p)\ (25 + 8\ p^2)$$

> expand(d);

$$25 + 8\ p^2 - 100\ p - 32\ p^3\ .$$

Разложение многочлена на множители осуществляется командой **factor(eq)**. *Пример:* зависимость полных издержек производства *K* от объема производства *x* выражается с помощью формулы: $K = x^3 - 4x^2 + 4x$.

> K:=x^3-4*x^2+4*x;

$$K := x^3 - 4\ x^2 + 4\ x$$

> **factor(K);**

$$x\ (x - 2)^2\ .$$

Команда **expand** может иметь дополнительный параметр, позволяющий при раскрытии скобок оставлять определенное выражение без изменений. *Например*, пусть требуется каждое слагаемое выражения $\ln x + e^x - y^2$ умножить на выражение (*x*+*a*).

Тогда в командной строке следует написать:

> expand((x+a)*(ln(x)+exp(x)-y^2), (x+a));

$$(x + a)\ln x + (x + a)e^x - (x + a)y^2$$

Дробь можно привести к нормальному виду с помощью команды **normal(eq)**. *Например:* пусть выпуск некоторого вида продукции описывается производственной функцией

> Q:=(K^4-L^4)/((K^2+L^2)*K*L);

$$Q := \frac{K^4 - L^4}{(K^2 + L^2)\,K\,L}$$

> normal(Q);

$$\frac{K^2 - L^2}{L\,K}.$$

Упрощение выражений осуществляется командой **simplify(eq)**.

Пример: эластичность спроса по цене задана выражением $E_p = \frac{p}{\left(p + \frac{1}{p}\right)} \cdot \frac{p - (p+1)}{p^2}$

> Ep:=(p/(1+1/p))*((p-(p+1))/p^2):
> simplify(Ep);

$$-\frac{1}{p+1}.$$

Приведение подобных членов в выражении осуществляется командой **collect(exp,var),** где **exp** – выражение, **var** – имя переменной, относительно которой следует собирать подобные. В команде **simplify** в качестве параметров можно указать, какие выражения преобразовывать. Например, при указании **simplify(eq,trig)** будет производиться упрощение при использовании большого числа тригонометрических соотношений. Стандартные параметры имеют названия: **power** – для степенных преобразований; **radical** или **sqrt** – для преобразования корней; **exp** – преобразование экспонент; **ln** – преобразование логарифмов. Использование параметров намного увеличивает эффективность команды **simplify**.

Объединить показатели степенных функций или понизить степень тригонометрических функций можно при помощи команды **combine(eq,param)**, где **eq** – выражение, **param** – параметры, указывающие, какой тип функций преобразовать, например, **trig** – для тригонометрических, **power** – для степенных. *Пример:* зависимость потребляемой на бытовые нужды города электроэнергии y (кВт.ч) от времени суток x (час) выражается следующей формулой: $y = a + b\cos\frac{\pi}{12}(x+3)$

> combine(a+b*cos(Pi*(x+3)/12), trig);

$$a + b\cos\left(\frac{1}{12}\pi\,x + \frac{1}{4}\pi\right).$$

Для упрощения выражений, содержащих не только квадратные корни, но и корни других степеней, лучше использовать команду **radnormal(eq)**. *Пример:* рыночная цена единицы продукции составит (ден.ед)

> sqrt(3+sqrt(3)+(10+6*sqrt(3))^(1/3))=
radnormal(sqrt(3+sqrt(3)+(10+6*sqrt(3))^(1/3)));

$$\sqrt{3+\sqrt{3}+(10+6\sqrt{3})^{1/3}} = 1+\sqrt{3}\,.$$

Все вычисления в *Maple* по умолчанию производятся символьно, то есть результат будет содержать в явном виде иррациональные константы, такие как, *e*, π и другие. Чтобы получить приближенное значение в виде числа с плавающей запятой, следует использовать команду **evalf(expr,t),** где **expr** – выражение, **t** – точность, выраженная в числах после запятой. *Например*, для предыдущего выражения рыночная цена единицы продукции составит:

>evalf(%);

$$3. = 3.$$

С помощью команды **convert(exp, param)**, где **exp** – выражение, которое будет преобразовано в указанный тип **param**. В частности, можно преобразовать выражение, содержащее sin*x* и cos*x*, в выражение, содержащее только tg*x*, если указать в качестве параметра **tan**, или, наоборот, tg*x*, ctg*x* можно перевести в sin*x* и cos*x*, если в параметрах указать **sincos**.

Вообще, команда **convert** имеет более широкое назначение. Она осуществляет преобразование выражения одного типа в другой. Например: **convert(list, vector)** – преобразование некоторого списка **list** в вектор с теми же элементами; **convert(expr, string)** – преобразование математического выражения в его текстовую запись. Для вызова подробной информации о назначении параметров команды **convert** следует обратиться к справочной системе, набрав **convert[termin]**.

Если вы забыли параметры какой-либо команды, то можно воспользоваться справочной системой *Maple*. Для вызова справки по конкретной команде, следует выделить набранное имя этой команды и нажать клавишу F1. Если команда набрана правильно, то появится описание этой команды (в большинстве версий *Maple* помощь на английском языке).

Контрольные вопросы.

1. Как перевести командную строку в текстовую и наоборот?
2. Как представляются в *Maple* основные математические константы?
3. Опишите виды представления рационального числа в *Maple*.
4. Как получить приближенное значение рационального числа?
5. Какими разделительными знаками заканчиваются команды в *Maple* и чем они отличаются?
6. Какой командой осуществляется вызов библиотеки подпрограмм?
7. Объясните назначение команд **factor**, **expand**, **normal**, **simplify**, **combine**, **convert**.

Глава II. Аналитические преобразования. Элементарная математика в экономических задачах

§1. Способы задания функций. Замена переменных

В Maple имеется несколько способов представления функции.

Способ 1. Определение функции с помощью оператора присваивания (:=): какому-то выражению присваивается имя, например: зависимость издержек производства C от объема выпускаемой продукции Q выражается функцией $C = 30Q - 0{,}08Q^3$

> C:=30*Q-0.08*Q^3;

$$C := 30Q - 0.08Q^3$$

Если задать конкретное значение переменной Q, то получится значение функции C для этого Q. *Например*, если продолжить предыдущий пример и вычислить значение C при $Q=10$ ед., то следует записать:

> Q:=10;

$$Q := 10$$

> C;

220.00

После выполнения этих команд переменная Q имеет заданное значение 10.

Чтобы насовсем не присваивать переменной конкретного значения, удобнее использовать команду подстановки **subs({x1=a1, x2=a2,…, },f),** где в фигурных скобках указываются переменные ***xi*** и их новые значения ***ai*** (*i*=1,2,…), которые следует подставить в функцию **f**. *Например*: функция затрат (издержек) имеет вид: $C = 2Q_1^2 + 2Q_1Q_2 + Q_2^2$, где Q_1, Q_2 обозначают объемы выпуска товаров

> C:=2*Q1^2+2*Q1*Q2+Q2^2;

$$C := 2Q1^2 + 2Q1Q2 + Q2^2$$

> subs({Q1=1000,Q2=800},C);

4240000

Способ 2. Определение функции с помощью функционального оператора, который ставит в соответствие набору переменных **(x1,x2,…)** одно или несколько выражений **(f1,f2,…)**. *Например,* определение функции двух переменных с помощью функционального оператора выглядит следующим образом:

>C(Q1,Q2):=2*Q1^2+2*Q1*Q2+Q2^2;

$$C(Q1, Q2) := 2Q1^2 + 2Q1Q2 + Q2^2$$

Обращение к этой функции осуществляется наиболее привычным в математике способом, когда в скобках вместо аргументов функции указываются конкретные значения переменных. В продолжение предыдущего *примера* вычисляется значение функции:

> C(1000,800);

4240000

Способ 3. С помощью команды **unapply(expr,x1,x2,…)**, где **expr** – выражение, **x1,x2,…** – набор переменных, от которых оно зависит, можно преобразовать выражение **expr** в функциональный оператор. *Например:*

> C:=unapply(2*Q1^2+2*Q1*Q2+Q2^2,Q1,Q2);

$$C := (Q1, Q2) -> 2Q1^2 + 2Q1Q2 + Q2^2$$

> C(1000,800);

4240000

В *Maple* имеется возможность определения неэлементарных функций вида

$$f(x)=\begin{cases} f_1(x),\ x<a_1 \\ f_2(x),\ a_1\le x<a_2 \\ \ldots\ldots\ldots\ldots \\ f_n(x),\ x\ge a_n \end{cases}$$

посредством команды

> piecewise(cond_1,f1, cond_2, f2, …).

Например, функция мощности y потребляемой городом электроэнергии задается формулой

$$y=\begin{cases} a, & t<6 \\ a+b\sin\dfrac{\pi}{18}(t-6), & t\ge 6, \end{cases}$$

где t – текущее время суток.

записывается следующим образом:

> y:=piecewise(t<6, a, t>=6,a+b*sin(Pi*(t-6)/18));

$$y:=\begin{cases} a & t<6 \\ a+b\sin\left(\dfrac{\pi\,(t-6)}{18}\right) & 6\le t \end{cases}$$

§2. Операции оценивания

Оценивание вещественных выражений.

В *Maple* имеются следующие команды оценивания вещественных выражений:

frac(expr) – вычисление дробной части выражения **expr**;

trunc(expr) – вычисление целой части выражения **expr**;

round(expr) – округление выражения **expr**;

Оценивание комплексных выражений.

Вещественную и мнимую части комплексного выражения $z=x+iy$ можно найти с помощью команд **Re(z)** и **Im(z)**. *Например:* коэффициент сглаживания при прогнозировании темпов инфляции $S=0.46+0.54i$

> S:=0.46+I*0.54:

> Re(S);Im(S);

0.46

0.54 .

Если $z=x+iy$, то комплексно сопряженное ему выражение $w=z^*=x-iy$ можно найти с помощью команды **conjugate(z)**. Продолжение предыдущего *примера*:

>w:=conjugate(S);

$w := 0.46 - 0.54\,I$

Модуль и аргумент комплексного выражения **z** можно найти с помощью команды **polar(z)**, которую необходимо предварительно вызвать из стандартной библиотеки командой **readlib**. *Например:* коэффициент сглаживания при прогнозировании темпов инфляции $S = i$

> readlib(polar): polar(I);

$$\mathrm{polar}\left(1, \frac{\pi}{2}\right)$$

В строке вывода в скобках через запятую указаны модуль числа *i*, равный единице и его аргумент, равный $\frac{\pi}{2}$. Если комплексное выражение очень сложное или содержит параметры, то команды **Re(z)** и **Im(z)** не дают требуемого результата. Получить вещественную и мнимую части комплексного выражения **z** можно, если использовать команду преобразования комплексных выражений **evalc(z)**. *Например:*

>z:=ln(1-I*sqrt(3))^2;

$$z := \ln(1 - \sqrt{3}\ I)^2$$

> evalc(Re(z)); evalc(Im(z));

$$\ln(2)^2 - \frac{\pi^2}{9}$$

$$-\frac{2}{3}\ln(2)\,\pi$$

§3. Решение уравнений

Решение обыкновенных уравнений.

Для решения уравнений в *Maple* существует универсальная команда **solve(eq,x)**, где **eq** – уравнение, **x** – переменная, относительно которой уравнение надо разрешить. В результате выполнения этой команды в строке вывода появится выражение, которое является решением данного уравнения. *Например:* спрос и предложение на некоторый товар на рынке описываются зависимостями вида $D(P) = 5 - 3P$, $S(P) = 4P + 1$. Определите равновесную цену.

> solve(5-3*P=4*P+1,P);

$$\frac{4}{7}$$

Если уравнение имеет несколько решений, которые вам понадобятся для дальнейших расчетов, то команде **solve** следует присвоить какое-нибудь имя **name**. Обращение к какому-либо **k**–ому решению данного уравнения производится указанием его имени с номером решения **k** в квадратных скобках: **name[k]**. *Например:* в течение года цена товара два раза увеличивалась на один и тот же процент. Первоначальная цена составляла 10 у.е. После второго повышения она составила 12,1 у.е. На сколько процентов повышалась цена

товара оба раза? Сколько процентов составила цена после второго повышения.

По условию задачи $10\left(1+\frac{x}{100}\right)\left(1+\frac{x}{100}\right)=12.1$

> y:=solve(10*(1+x/100)*(1+x/100)=12.1,x);

$y := 10., -210.$

> y[1];

10.

> y[2];

-210.

Следовательно, цена товара увеличивалась каждый раз на 10 %.

> 100+2*y[1];

120.

Таким образом, цена после второго повышения составила 120%.

Решение систем уравнений.

Системы уравнений решаются с помощью такой же команды **solve({eq1,eq2,…},{x1,x2,…})**, только теперь в параметрах команды следует указывать в первых фигурных скобках через запятую уравнения, а во вторых фигурных скобках перечисляются через запятую переменные, относительно которых требуется решить систему. Если вам будет необходимо для дальнейших вычислений использовать полученные решения уравнений, то команде **solve** следует присвоить какое-нибудь имя **name**. Затем выполняется присвоения команда **assign(name)**. После этого над решениями можно будет производить математические операции. *Например:* автозавод выпустил автомобили двух марок (*A* и *B*) в количестве 52 000 штук. На следующий год запланировано увеличение выпуска автомобилей марки *A* на 75 %, а марки *B* – на 140 %. В результате выпуск автомобилей должен увеличиться в два раза. Сколько автомобилей каждой марки автозавод выпустил в текущем году и выпустит на следующий год? Сколько всего выпустит автомобилей автозавод.

Обозначим через *x* количество автомобилей марки *A*, выпущенных заводом в текущем году, а *y* – марки *B*. По условию задачи

> **h:=solve({x+y=52000,1.75*x+2.4*y=104000},{x,y});**

$h := \{x = 32000., y = 20000.\}$

> **assign(h); simplify(x+y);**

52000.

Численное решение уравнений.

Для численного решения уравнений, в тех случаях, когда трансцендентные уравнения не имеют аналитических решений, используется специальная команда **fsolve(eq,x)**, параметры которой такие же, как и команды **solve**. *Например:*

> **x:=fsolve(cos(t)=t,t);**

$x := .7390851332$

Решение тригонометрических уравнений.

Команда **solve**, примененная для решения тригонометрического уравнения, выдает только главные решения, то есть решения в интервале $[0,2\pi]$. Для того, чтобы получить все решения, следует предварительно ввести дополнительную команду **_EnvAllSolutions:=true.** *Например:*

> **_EnvAllSolutions:=true:**

> **solve(sin(t)=cos(t),t);**

$$\frac{1}{4}\pi + \pi \_Z1\sim$$

В *Maple* символ *_Z~* обозначает константу целого типа, поэтому решение данного уравнения в привычной форме имеет вид $x := \pi / 4 + \pi n$, где n – целые числа.

§4. Решение неравенств

Команда solve применяется также для решения неравенств. Решение неравенства выдается в виде интервала изменения искомой переменной. В том случае, если решение неравенства полуось, то в поле вывода появляется конструкция вида RealRange(–∞, Open(a)), которая означает, что $x \in (-\infty, a)$, a – некоторое число. Слово Open означает, что интервал с открытой границей. Если этого слова нет, то соответствующая граница интервала включена во множество решений.

Например: Для одного из предприятий-монополистов зависимость объема спроса на продукцию q (единиц в месяц) от ее цены p (тыс.руб.) задается формулой $q = 150 - 10p$. Определите максимальный уровень цены p (тыс.руб.), при котором значение выручки предприятия за месяц $r = q \cdot p$ составит не менее 440 тыс.руб.

> solve((150-10*p)*p>=440,p);

$$\mathrm{RealRange}(4, 11)$$

Итак, максимальный уровень цены 11 тыс.рублей.

Если вы хотите получить решение неравенства не в виде интервального множества типа $x \in (a, b)$, а в виде ограничений для искомой переменной типа $a<x$, $x< b$, то переменную, относительно которой следует разрешить неравенство, следует указывать в фигурных скобках. *Например:*

> solve((150-10*p)*p>=440,{p});

$$\{4 \le p, p \le 11\}$$

Решение систем неравенств.

С помощью команды **solve** можно также решить систему неравенств.

Контрольные вопросы.

1. Опишите способы задания функций в *Maple*.
2. Какие операции оценивания производятся в *Maple* с действительными выражениями?
3. Для чего предназначена команда **evalf**?

4. С помощью каких команд можно найти вещественную и мнимую части комплексного выражения, а также его модуль и аргумент, и комплексно сопряженное ему число? Какую роль выполняет команда **evalc**?
5. Для чего предназначена команда **solve**?
6. Какие команды используются для численного решения уравнений и для решения рекуррентных уравнений?
7. В каком виде выдается решение неравенства? Как отличить в строке вывода закрытый интервал от открытого?

Глава III. Построение графиков

§1. Двумерные графики

Команда plot и ее параметры.

Для построения графиков функции **f(x)** одной переменной (в интервале $a \le x \le b$ по оси Ox и в интервале $c \le y \le d$ по оси Oy) используется команда **plot(f(x), x=a..b, y=c..d, parameters),** где **parameters** – параметры управления изображением. Если их не указывать, то будут использованы установки по умолчанию. Настройка изображения также может осуществляться с панели инструментов.

Основные параметры команды **plot**:

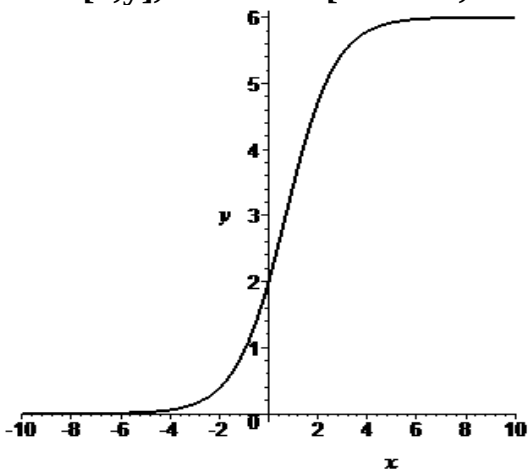
1) **title=”text”,** где **text**-заголовок рисунка (текст можно оставлять без кавычек, если он содержит только латинские буквы без пробелов).
2) **coords=polar –** установка полярных координат (по умолчанию установлены декартовы).
3) **axes** – установка типа координатных осей: **axes=NORMAL** – обычные оси; **axes=BOXED** – график в рамке со шкалой;
axes=FRAME – оси с центром в левом нижнем углу рисунка;
axes=NONE – без осей.
4) **scaling** – установка масштаба рисунка: **scaling=CONSTRAINED** – одинаковый масштаб по осям; **scaling=UNCONSTRAINED** – график масштабируется по размерам окна.
5) **style=LINE(POINT)** – вывод линиями (или точками).
6) **numpoints=n** – число вычисляемых точек графика (по умолчанию **n=49**).
7) **color** – установка цвета линии: английское название цвета, например, **yellow** – желтый и т.д.
8) **xtickmarks=nx** и **ytickmarks=ny** – число меток по оси Ox и оси Oy, соответственно.
9) **thickness=n,** где **n=1,2,3…** - толщина линии (по умолчанию **n=1**).
10) **linestyle=n** – тип линии: непрерывная, пунктирная и т.д. (**n=1** – непрерывная, установлено по умолчанию).
11) **symbol=s** – тип символа, которым помечают точки: **BOX, CROSS, CIRCLE, POINT, DIAMOND**.

12) **font=[f,style,size]** – установка типа шрифта для вывода текста: **f** задает название шрифтов: **TIMES, COURIER, HELVETICA, SYMBOL; style** задает стиль шрифта: **BOLD, ITALIC, UNDERLINE; size** – размер шрифта в pt.
13) **labels=[tx,ty]** – надписи по осям координат: **tx** – по оси *Ox* и **ty** – по оси *Oy*.
14) **discont=true** – указание для построения бесконечных разрывов.

С помощью команды **plot** можно строить помимо графиков функций **y=f(x),** заданной явно, также графики функций, заданных параметрически **y=y(t), x=x(t)**, если записать команду **plot([y=y(t), x=x(t), t=a..b], parameters)**.

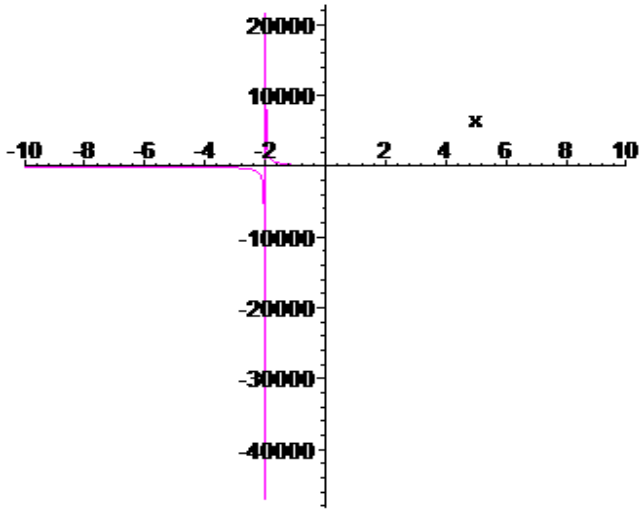
Пример. Построить график функции предметов потребления (логистическая кривая) $y = \frac{6}{1+2e^{-x}}$ жирной линией. Набираем:

>plot(6/(1+2*exp(-x)),labels=[x,y],labelfont=[TIMES,ITALIC,12],thickness=2);



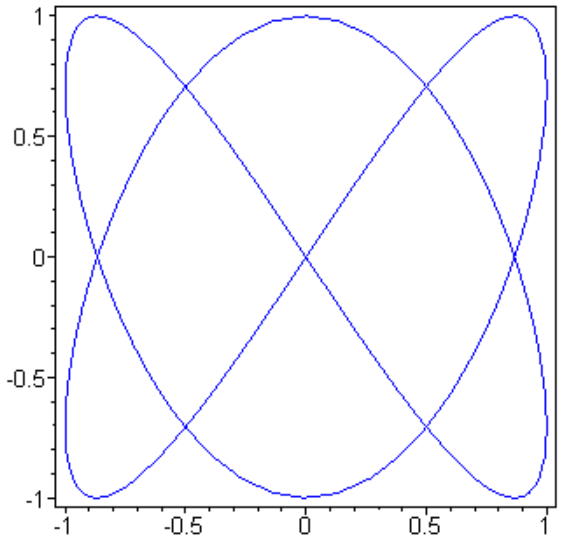
Построить график разрывной функции спроса на товар $y = \frac{200}{x+2}$, где x – цена товара

> plot(200/(x+2),color=magenta);



Построить график параметрической кривой y = sin2t , x = cos3t, $0 \le t \le 2\pi$ в рамке. Наберите:

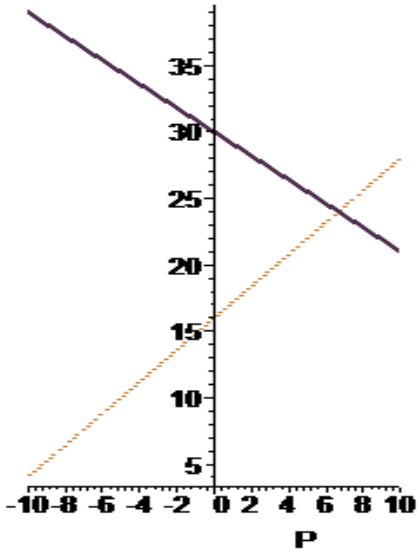
> **plot([sin(2*t),cos(3*t),t=0..2*Pi], axes=BOXED, color=blue);**



Построить два графика на одном рисунке: график функции долговременного спроса D и предложения S от цены P на мировом рынке нефти имеют, соответственно, вид $D(P) = 30 - 0{,}9P$, $S(P) = 16 + 1{,}2P$. Определить графически точку равновесной цены.

Наберите:

> **plot([30-0.9*P, 16+1.2*P],scaling=CONSTRAINED, color=[violet,gold], linestyle=[1,2], thickness=[3,2]);**



Построение графика функции, заданной неявно.

Функция задана неявно, если она задана уравнением F(x, y) = 0 . Для построения графика неявной функции используется команда implicitplot из графического пакета plots.

Вывод текстовых комментариев на рисунок.

В пакете **plots** имеется команда **textplot** для вывода текстовых комментариев на рисунок: **textplot([xo,yo,'text'], options),** где **xo, yo** – координаты точки, с которой начинается вывод текста **'text'**.

Вывод нескольких графических объектов на один рисунок.

Часто бывает необходимо совместить на одном рисунке несколько графических объектов, полученных при помощи различных команд, например, добавить графику, нарисованному командой **plot**, текстовые надписи, полученные командой **textplot**. Для этого результат действия команды присваивается некоторой переменной:

> **p:=plot(…): t:=textplot(…):**

При этом на экран вывод не производится. Для вывода графических изображений необходимо выполнить команду из пакета **plots:**

> **with(plots): display([p,t], options)**.

Построение двумерной области, заданной неравенствами.

Если необходимо построить двумерную область, заданную системой неравенств $f1(x, y) > c1$, $f2(x, y) > c2$,..., $fn(x, y) > cn$, то для этого можно использовать команду **inequal** из пакета **plots**. В команде **inequals({f1(x,y)>c1,…,fn(x,y)>cn}, x=x1…x2, y=y1..y2, options)** в фигурных скобках указывается система неравенств, определяющих область, затем размеры координатных осей и параметры. Параметры регулируют цвета открытых и закрытых границ, цвета внешней и внутренней областей, а также толщину линий границ:

– **optionsfeasible=(color=red)** – установка цвета внутренней области;

– **optionsexcluded=(color=yellow)** – установка цвета внешней области;

– **optionsopen(color=blue, thickness=2)** – установка цвета и толщины линии открытой границы;

– **optionsclosed(color=green,thickness=3)** – установка цвета и толщины линии закрытой границы.

§2. Трехмерные графики. Анимация

График поверхности, заданной явной функцией. График функции z = f (x, y) можно нарисовать, используя команду plot3d(f(x,y), x=x1…x2, y=y1…y2, options). Параметры этой команды частично совпадают с параметрами команды plot. К часто используемым параметрам команды plot3d относится light=[angl1, angl2, c1, c2, c3] – задание подсветки поверхности, создаваемой источником света из точки со сферическими координатами (angl1, angl2). Цвет определяется долями красного (c1), зеленого (c2) и синего (c3) цветов, которые находятся в интервале [0,1]. Параметр style=opt задает стиль рисунка: POINT –точки, LINE – линии, HIDDEN – сетка с удалением невидимых линий, PATCH – заполнитель

(установлен по умолчанию), WIREFRAME – сетка с выводом невидимых линий, CONTOUR – линии уровня, PATCHCONTOUR – заполнитель и линии уровня.

Параметр shading=opt задает функцию интенсивности заполнителя, его значение равно xyz – по умолчанию, NONE – без раскраски.

График поверхности, заданной параметрически.

Если требуется построить поверхность, заданную параметрически: $x=x(u,v)$, $y=y(u,v)$, $z=z(u,v)$, то эти функции перечисляются в квадратных скобках в команде: **plot3d([x(u,v), y(u,v), z(u,v)], u=u1..u2, v=v1..v2)**.

График поверхности, заданной неявно.

Трехмерный график поверхности, заданной неявно уравнением $F(x, y, z) = c$, строится с помощью команды пакета **plot: implicitplot3d(F(x,y,z)=c, x=x1..x2, y=y1..y2, z=z1..z2),** где указывается уравнение поверхности $F(x, y, z) = c$ и размеры рисунка по координатным осям.

График пространственных кривых.

В пакете **plot** имеется команда **spacecurve** для построения пространственной кривой, заданной параметрически: $x = x(t)$, $y = y(t)$, $z = z(t)$. Параметры команды**:** > **spacecurve([x(t),y(t),z(t)],t=t1..t2)**, где переменная **t** изменяется от **t1** до **t2**.

Анимация.

Maple позволяет выводить на экран движущиеся изображения с помощью команд **animate** (двумерные) и **animate3d** (трехмерные) из пакета **plot**. Среди параметров команды **animate3d** есть **frames** – число кадров анимации (по умолчанию **frames=8**).

Трехмерные изображения удобнее настраивать не при помощи опций команды **plot3d**, а используя контекстное меню программы. Для этого следует щелкнуть правой кнопкой мыши по изображению. Тогда появится контекстное меню настройки изображения. Команды этого меню позволяют изменять цвет изображения, режимы подсветки, устанавливать нужный тип осей, тип линий и управлять движущимся изображением.

Контекстное меню настройки изображения:

Контрольные вопросы.

1.С помощью каких команд строятся графики на плоскости и в пространстве? Какие аргументы имеют эти команды?

2. С помощью какой команды можно построить график неявной функции? Опишите ее параметры.
3. Для чего предназначена команда **display**?
4. Какая команда позволяет построить двумерную область, заданную системой неравенств?
5. С помощью какой команды можно построить график пространственной кривой?
6. Какие возможности предоставляют команды **animate** и **animate3d**?

Глава IV. Математический анализ в экономической деятельности

§1. Вычисление пределов

В Maple для некоторых математических операций существует по две команды: одна прямого, а другая – отложенного исполнения. Имена команд состоят из одинаковых букв за исключением первой: команды прямого исполнения начинаются со строчной буквы, а команды отложенного исполнения – с заглавной. После обращения к команде отложенного действия математические операции (интеграл, предел, производная и т.д.) выводятся на экран в виде стандартной аналитической записи этой операции. Вычисление в этом случае сразу не производится. Команда прямого исполнения выдает результат сразу.

Для вычисления пределов имеются две команды:

1) прямого исполнения – limit(expr,x=a,par), где expr – выражение, предел которого следует найти, a – значение точки, для которой вычисляется предел, par – необязательный параметр для поиска односторонних пределов (left – слева, right – справа) или указание типа переменной (real – действительная, complex – комплексная).

2) отложенного исполнения – Limit(expr,x=a,par), где параметры команды такие же, как и в предыдущем случае. *Пример* действий этих команд: что произойдет со спросом на товар, если фирма решит резко поднять на него цену ($x \to \infty$)? Спрос на который выражается функцией $y = \frac{400}{x+2}$, где x – цена товара.

> **Limit(400/(x +2),x= infinity);**

$$\lim_{x \to \infty} \frac{400}{x+2}$$

> **limit(400/(x +2),x= infinity);**

$$0$$

Задача показала, что при неограниченном росте цен спрос приближается к нулю.

С помощью этих двух команд принято записывать математические выкладки в стандартном аналитическом виде, *например:*

> **Limit(400/(x +2),x= infinity)= limit(400/(x +2),x= infinity);**

$$\lim_{x\to\infty}\frac{400}{x+2}=0$$

Односторонние пределы вычисляются с указанием параметров: left – для нахождения предела слева и righ – справа. *Например:*

> **Limit(1/(1+exp(1/t)),t=0,left)= limit(1/(1+exp(1/t)),t=0,left);**

$$\lim_{t\to 0-}\frac{1}{1+\mathbf{e}^{\left(\frac{1}{t}\right)}}=1$$

§2. Дифференцирование
Вычисление производных.

Для вычисления производных в *Maple* имеются две команды:

1) прямого исполнения – **diff(f,x)**, где **f** – функция, которую следует продифференцировать, **x** – имя переменной, по которой производится дифференцирование.

2) отложенного исполнения – **Diff(f,x)**, где параметры команды такие же, как и в предыдущей. Действие этой команды сводится к аналитической записи производной в виде $\frac{\partial}{\partial x}f(x)$. После выполнения дифференцирования, полученное выражение желательно упростить. Для этого следует использовать команды **simplify factor** или **expand**, в зависимости от того, в каком виде вам нужен результат.

Пример: определите выражение, описывающее производительность труда, если объем продукции u, произведенный бригадой рабочих, описан уравнением $u=-\frac{5}{6}t^3+\frac{15}{2}t^2+100t+50$ (ед.).

Производительность труда выражается производной (ед.ч.)

>Diff((-5/6)*t^3+(15/2)*t^2+100*t+50,t)=diff(((-5/6)*t^3+(15/2)*t^2+100*t+50,t);

$$\frac{d}{dt}\left(-\frac{5}{6}t^3+\frac{15}{2}t^2+100\,t+50\right)=-\frac{5}{2}t^2+15\,t+100$$

Для вычисления производных старших порядков следует указать в параметрах x$n, где n – порядок производной; *например:* (условие предыдущей задачи) определите выражение, описывающее скорость труда (ед.ч.^{2})

>Diff((-5/6)*t^3+(15/2)*t^2+100*t+50,t$2)=diff(((-5/6)*t^3+(15/2)*t^2+100*t+50,t$2);

$$\frac{d^2}{dt^2}\left(-\frac{5}{6}t^3+\frac{15}{2}t^2+100\,t+50\right)=-5\,t+15$$

.

§3. Исследование функции
Экстремумы. Наибольшее и наименьшее значение функции.

В Maple для исследования функции на экстремум имеется команда extrema(f,{cond},x,'s') , где f - функция, экстремумы которой ищутся, в фигурных скобках {cond} указываются ограничения для переменной, x – имя переменной, по которой ищется экстремум, в апострофах 's' – указывается имя переменной, которой будет присвоена координата точки экстремума. Если оставить пустыми фигурные скобки {}, то поиск экстремумов будет производиться на всей числовой оси. Результат действия этой команды относится к типу set. *Пример:* найти максимум прибыли, если издержки и доход определяются следующими формулами: $C(Q) = Q^3 - 37Q^2 + 169Q + 4000$, $R(Q) = 100Q - Q^2$.

Прибыль определяется как: $\Pi(Q) = R(Q) - C(Q) = -Q^3 + 36Q^2 - 69Q - 4000$

> **eadlib(extrema):extrema(-Q^3+36*Q^2-69*Q-4000,{},Q,'Q0');Q0;**

$$\{-4034, 1290\}$$

$$\{\{Q = 1\}, \{Q = 23\}\}$$

Таким максимальная прибыль достигается при $Q = 23$, $\Pi_{\max} = 1290$.

В первой строке вывода приводится экстремум функции, а во второй строке вывода – точка этого экстремума.

Но эта команда не может дать ответ на вопрос, какая из точек экстремума есть максимум, а какая – минимум. Для нахождения максимума функции f(x) по переменной x на интервале x ∈ [x1, x2] используется команда maximize(f,x,x=x1..x2), а для нахождения минимума функции f(x) по переменной x на интервале x ∈ [x1, x2] используется команда minimize(f, x, x=x1..x2).

Если после переменной указать **'infinity'** или интервал **x=-nfinity..+infinity**, то команды **maximize** и **minimize** будут искать, соответственно, максимумы и минимумы на всей числовой оси как во множестве вещественных чисел, так и комплексных. Если такие параметры не указывать, то поиск максимумов и минимумов будет производиться только во множестве вещественных чисел.

Координаты точек максимума или минимума можно получить, если в параметрах этих команд после переменной записать через запятую новую опцию **location**. В результате в строке вывода после самого максимума (минимума) функции будут в фигурных скобках указаны координаты точек максимума (минимума). В строке вывода получились координаты минимумов и значения функции в этих точках.

§4. Интегрирование

Аналитическое и численное интегрирование.

Неопределенный интеграл $\int f(x)dx$ вычисляется с помощью 2-хкоманд: 1) прямого исполнения – **int(f, x),** где **f** – подынтегральная функция, **x** – переменная интегрирования; 2) отложенного исполнения – **Int(f, x)** – где

параметры команды такие же, как и в команде прямого исполнения **int**. Команда **Int** выдает на экран интеграл в аналитическом виде математической формулы.

Для вычисления определенного интеграла $\int_a^b f(x)dx$ в командах int и Int добавляются пределы интегрирования, *например*: стоимость перевозки 1 тонны на 1 км – *y* руб/км (тариф на перевозки) убывает в зависимости от расстояния и определяется по следующей формуле: $y = \frac{a}{x+b}$. Найти зависимость суммарной стоимости перевозки 1 тонны груза от пройденного пути. Пусть A – искомая стоимость. Тогда

> A:=int(a/(x+b),x);

$$A := a \ln(x + b)$$

Обобщим задачу.

Пример: найти стоимость перевозки *M* тонн груза по железной дороге на расстояние *L* км при условии, что тариф *y* перевозки одной тонны убывает на *a* руб. на каждом последующем километре.

Решение задачи сводится к вычислению интеграла

>A:=Int(M*(y-a*x),x=0..L)=int(M*(y-a*x),x=0..L);

$$A := \int_0^L M\,(y - a\,x)\,dx = -\frac{M\,a\,L^2}{2} + M\,y\,L\ .$$

Если в команде интегрирования добавить опцию continuous: int(f, x, continuous), то Maple будет игнорировать любые возможные разрывы подынтегральной функции в диапазоне интегрирования. Это позволяет вычислять несобственные интегралы от неограниченных функций. Несобственные интегралы с бесконечными пределами интегрирования вычисляются, если в параметрах команды int указывать, например, x=0..+infinity. Численное интегрирование выполняется командой evalf(int(f, x=x1..x2), e), где e – точность вычислений (число знаков после запятой).

Интегралы, зависящие от параметра. Ограничения для параметров. Если требуется вычислить интеграл, зависящий от параметра, то его значение может зависеть от знака этого параметра или каких-либо других ограничений. Рассмотрим в качестве примера интеграл $\int_0^{+\infty} e^{-ax}dx$, который, как известно из математического анализа, сходится при a>0 и расходится при a<0. Если вычислить его сразу, то получится:

> Int(exp(-a*t),t=0..+infinity)=int(exp(-a*t),t=0..+infinity);

$$\int_0^{\infty} \mathbf{e}^{(-a\,t)}\,dt = \lim_{t \to \infty} -\frac{\mathbf{e}^{(-a\,t)} - 1}{a}$$

Таким способом интеграл с параметром не вычислить. Для получения явного аналитического результата вычислений следует сделать какие-либо

предположения о значении параметров, то есть наложить на них ограничения. Это можно сделать при помощи команды **assume(expr1)**, где **expr1** – неравенство. Дополнительные ограничения вводятся с помощью команды **additionally(expr2)**, где **expr2** – другое неравенство, ограничивающее значение параметра с другой стороны. После наложения ограничений на параметр *Maple* добавляет к его имени символ (~), например параметр **a**, на который были наложены некоторые ограничения, в сроке вывода будет иметь вид: *a~*.
Описание наложенных ограничений параметра a можно вызвать командой about(a). Вернемся к вычислению интеграла с параметром $\int_0^{+\infty} e^{-ax}dx$ которое следует производить в таком порядке:

> **assume(a>0);**

> **Int(exp(-a*t),t=0..+infinity)=int(exp(-a*t),t=0..+infinity);**

$$\int_0^{\infty} \mathbf{e}^{(-a\sim t)}\, dt = \frac{1}{a\sim}$$

Обучение основным методам интегрирования.

В Maple имеется пакет student, предназначенный для обучения математике. Он содержит набор подпрограмм, предназначенных для выполнения расчетов шаг за шагом, так, чтобы была понятна последовательность действий, приводящих к результату. К таким командам относятся интегрирование по частям inparts и замена переменной changevar.

Формула интегрирования по частям: ∫u(x)v'(x)dx = u(x)v(x) −∫u'(x)v(x)dx

Если обозначить подынтегральную функцию f=u(x)v’(x), то параметры команды интегрирования по частям такие: intparts(Int(f, x), u), где u – именно та функция u(x), производную от которой предстоит вычислить по формуле интегрирования по частям. Если в интеграле требуется сделать замену переменных x=g(t) или t=h(x), то параметры команды замены переменных такие: changevar(h(x)=t, Int(f, x), t), где t − новая переменная. Обе команды intparts и changevar не вычисляют окончательно интеграл, а лишь производят промежуточную выкладку. Для того, чтобы получить окончательный ответ, следует, после выполнения этих команд ввести команду value(%); где % - обозначают предыдущую строку. Не забудьте, перед использованием описанных здесь команд обязательно загрузить пакет student командой with(student).

Пример. Полностью проделать все этапы вычисления интеграла по частям, для нахождения объема выпускаемой продукции за пять лет, если в функции Кобба-Дугласа имеет вид $f(t) = e^t(t+1)(100-3t)$

> **restart; with(student): Q=Int(exp(t)*(t+1)*(100-3*t),t=0..5);**

$$Q = \int_0^5 \mathbf{e}^{t}\,(t+1)\,(100-3\,t)\,dt$$

> **Q=intparts(Int(exp(t)*(t+1)*(100-3*t),t=0..5),(t+1)*(100-3*t));**

$$Q = 510\ \mathbf{e}^5 - 100 - \int_0^5 (97 - 6\,t)\,\mathbf{e}^t\,dt$$

> **intparts(%,97-6*t);**

$$Q = 443\ \mathbf{e}^5 - 3 + \int_0^5 -6\,\mathbf{e}^t\,dt$$

> **value(%);**

$$Q = 437\ \mathbf{e}^5 + 3$$

> **evalf(%);**

$$Q = 64859.55053$$

Итак, объем выпускаемой продукции за пять лет составит около 64859.

§5. Дифференциальное исчисление функций многих переменных

Частные производные.

Для вычисления частных производных функции $f(x1,\ldots, xm)$ используется уже хорошо известная вам команда **diff**. В этом случае эта команда имеет такой формат: **diff(f,x1$n1,x2$n2,…, xm$nm),** где **x1,…, xm** – переменные, по которым производится дифференцирование, а после знака **$** указаны соответствующие порядки дифференцирования. *Например*, частная производная $\frac{\partial^2 f}{\partial x \partial y}$ записывается в виде: **diff(f,x,y).**

Пример. Найти все частные производные 2-го порядка функции $f(x,y) = \frac{x-y}{x+y}$.

> **restart; f:=(x-y)/(x+y):**

> **Diff(f,x$2)=simplify(diff(f,x$2));**

$$\frac{\partial^2}{\partial x^2}\left(\frac{x-y}{x+y}\right) = -\frac{4\,y}{(x+y)^3}$$

> **Diff(f,y$2)=simplify(diff(f,y$2));**

$$\frac{\partial^2}{\partial y^2}\left(\frac{x-y}{x+y}\right) = \frac{4\,x}{(x+y)^3}$$

> **Diff(f,x,y)=diff(f,x,y);**

$$\frac{\partial^2}{\partial y\,\partial x}\left(\frac{x-y}{x+y}\right) = \frac{2\,(x-y)}{(x+y)^3}$$

Локальные и условные экстремумы функций многих переменных.

Для исследования функции на локальный и условный экстремум используется команда из стандартной библиотеки extrema(f,{cond},{x,y,…},'s'), где cond – ограничения для поиска условного экстремума, которые записываются в виде равенств.

После ограничений в фигурных скобках указываются все переменные, от которых зависит функция f, а затем в кавычках записывается s – имя

переменной, которой будут присвоены координаты точек экстремума. Если ограничений не указывать, то будет производиться поиск локального экстремума.

Но, команда extrema выдает все критические точки, то есть и те, в которых экстремума нет. Отсеять недающие экстремума критические точки можно с помощью непосредственной подстановки этих точек в функцию, например, оператором subs.

Пример. найти максимальную прибыль при использовании ресурсов. Функция прибыли имеет вид $П(x,y)=30\sqrt{x}\sqrt[3]{y}-5x-10y$ ($x-$ количество единиц первого ресурса, $y-$ второго).

> **restart: readlib(extrema):**
> **П:=30*sqrt(x)*(y)^(1/3)-5*x-10*y;**
> **extrema(П,{},{x,y},'s');s;**

$$П := 30\sqrt{x}\,y^{(1/3)} - 5\,x - 10\,y$$
$$\{\,135\,\}$$
$$\{\{x = 81, y = 27\}\}$$

Получилось один экстремум, поэтому максимальная прибыль равна 135 (ден.ед.), максимум достигается в точке (81,27).

§6. Интегральное исчисление функций многих переменных

В *Maple* имеются две специальные команды для вычисления двойных и тройных интегралов, содержащиеся в библиотеке **student**. Для вычисления двойных интегралов $\iint\limits_D f(x,y)dxdy$ используется команда **Doubleint(f(x, y), D),** где **D** – область интегрирования, записываемая в одном из следующих форматов:

1). **x=x1..x2, y=y1..y2**, где числа **x1, x2, y1, y2** задают прямоугольную область интегрирования;

2). **x=f1(y)..f2(y), y=y1..y2**, где **f1(y), f2(y)** − линии, ограничивающие область интегрирования слева и справа на
интервале от **y1** до **y2**;

3). **x=x1..x2, y=g1(x)..g2(x)** , где **g1(y), g2(y)** − линии, ограничивающие область интегрирования снизу и сверху на интервале от **x1** до **x2**.

Для вычисления тройных интегралов $\iiint\limits_V f(x,y,z)dxdydz$ используется команда **Tripleint(f(x, y, z),x, y, z, V),** где **V** – область интегрирования. Обе эти команды являются командами отложенного действия.

Чтобы получить значение интеграла, следует использовать команду **value(%)**.

value(%).

Повторные интегралы можно вычислять с помощью повторения команды int, *например*, повторный интеграл $\int_0^2 dy \int_0^1 x^2 y^3 dx$ вычисляется командой

> **int(int(x^2*y^3, x=0..1), y=0..2);**

$$\frac{4}{3}.$$

Пример. Вычислить двойной интеграл

$$\iint_D \sin(x+2y)dxdy$$

по области, ограниченной линиями $y=0, y=x, x+y=\frac{\pi}{2}$.

Замечание: сначала следует описать область интегрирования D в виде неравенств.

> **restart: with(student):**

> **J:=Doubleint(sin(x+2*y), x=y..Pi/2-y, y=0..Pi/2);**

$$J := \int_0^{\frac{\pi}{2}} \int_y^{\frac{\pi}{2}-y} \sin(x+2\,y)\,dx\,dy$$

> **J:=value(%);**

$$J := \frac{2}{3}.$$

§7. Ряды и произведения
Вычисление суммы ряда и произведений.

Конечные и бесконечные суммы $\sum_{n=a}^{b} S(n)$ вычисляются командой прямого исполнения **sum** и отложенного исполнения **Sum**. Аргументы этих команд одинаковые: **sum(expr, n=*a..b*),** где **expr** –выражение, зависящее от индекса суммирования, ***a..b*** – пределы индекса суммирования, указывающие, что суммировать следует от **n=*a*** до **n=*b***. Если требуется вычислить сумму бесконечного ряда, то в качестве верхнего предела вводится **infinity**.

Аналогичным образом вычисляются произведения $\prod_{n=a}^{b} P(n)$ командами прямого **product(P(n),n=a..b)** и отложенного действий **Product P(n),n=a..b).**

Пример: найти полную и *N*- частичную суммы ряда, общий член которого равен: $a_n = \frac{1}{(3n-2)(3n+1)}$

>restart; a[n]:=1/((3*n-2)*(3*n+1));

$$a_n := \frac{1}{(3\,n-2)\,(3\,n+1)}$$

>S[N]:=Sum(a[n],n=1..N)=sum(a[n],n=1..N);

$$S_N := \sum_{n=1}^{N} \frac{1}{(3n-2)(3n+1)} = -\frac{1}{3(3N+1)} + \frac{1}{3}$$

>S:=limit(rhs(S[N]),N=+infinity);

$$S := \frac{1}{3}.$$

Разложение функции в степенной ряд и ряд Тейлора.

Разложение функции f(x) в степенной ряд в окрестности точки *a*

$$f(x) = C_0 + C_1(x-a) + C_2(x-a)^2 + \ldots$$

в окрестности которой производится разложение, n – число членов ряда. Аналогичного действия команда taylor(f(x), x=a, n) раскладывает функции f(x) в окрестности точки x=a до порядка n-1 по формуле Тейлора.

Команды series и taylor выдают результат, имеющий тип series. Для того, чтобы иметь возможность дальнейшей работы с полученным разложением, его следует преобразовать в полином с помощью команды convert(%,polynom).

Функцию многих переменных f(x1,…,xn) можно разложить в ряд Тейлора по набору переменных (x1,…,xn) в окрестности точки (a1,…,an) до порядка n с помощью команды mtaylor(f(x), [x1,…,xn], n). Эта команда находится в стандартной библиотеке, поэтому перед использованием должна быть вызвана readlib(mtaylor).

Пример. 1. Разложить в степенной ряд $f(x) = e^{-x}\sqrt{x+1}$ в окрестности $x_0 = 0$ удерживая 5 первых членов.

> f(x)=series(exp(-x)*sqrt(x+1),x=0,5);

$$\mathrm{f}(x) = 1 - \frac{1}{2}x - \frac{1}{8}x^2 + \frac{13}{48}x^3 - \frac{79}{384}x^4 + \mathrm{O}(x^5)$$

2. Разложить $f(x,y) = \sin(x^2 + y^2)$ в ряд Тейлора в окрестности точки (0,0) до 6-го порядка

>readlib(mtaylor):

>f=mtaylor(sin(x^2+y^2),[x=0,y=0],7);

$$f = x^2 + y^2 - \frac{1}{6}x^6 - \frac{1}{2}y^2x^4 - \frac{1}{2}y^4x^2 - \frac{1}{6}y^6.$$

§8. Дифференциальные уравнения

Общее решение дифференциальных уравнений.

Для нахождения аналитических решений дифференциальных уравнений в Maple применяется команда dsolve(eq,var,options), где eq – дифференциальное уравнение, var – неизвестные функции, options – параметры.

Параметры могут указывать метод решения задачи, например, по умолчанию ищется аналитическое решение: type=exact. При составлении дифференциальных уравнений для обозначения производной применяется команда diff, *например*, дифференциальное уравнение $y''+y=x$ записывается в виде: diff(y(x),x$2)+y(x)=x.

Общее решение дифференциального уравнения зависит от произвольных постоянных, число которых равно порядку дифференциального уравнения. В Maple такие постоянные, как правило, обозначаются как _C1, _C2, и т.д.

Команда **dsolve** выдает решение дифференциального уравнения в невычисляемом формате. Для того, чтобы с решением можно было бы работать далее (например, построить график решения) следует отделить правую часть полученного решения командой **rhs(%)**.

Пример. Найти общее решение дифференциального уравнения y'+ycosx=sinxcosx.

> restart;

> **de:=diff(y(x),x)+y(x)*cos(x)=sin(x)*cos(x);**

$$de := \left(\frac{d}{dx}\,y(x)\right) + y(x)\cos(x) = \sin(x)\cos(x)$$

> **dsolve(de,y(x));**

$$y(x) = \sin(x) - 1 + \mathbf{e}^{(-\sin(x))}\,\_C1$$

Решение задачи Коши или краевой задачи.

Команда **dsolve** может найти решение задачи Коши или краевой задачи, если помимо дифференциального уравнения задать начальные или краевые условия для неизвестной функции. Для обозначения производных в начальных или краевых условиях используется дифференциальный оператор D, например, условие y''(0)=2 следует записать в виде (D@@2)(y)(0) = 2 , или условие y'(1)=0: D(y)(1) = 0 . Напомним, что производная n-го порядка записывается в виде (D@@n)(y) .

Пример. Изменение численности населения горнорудного поселка с течением времени описывается уравнением: $y' = 0.3y(2-10^{-4}y)$, где $y = y(t)$, t – время (лет). В начальный момент времени составляло 500 человек. Каким оно станет через три года?

> **de:=diff(y(x),x)=0,3*y(x)*(2-10^(-4)*y(x));**

$$de := \frac{d}{dx}\,y(x) = 0.3\,y(x)\left(2 - \frac{1}{10000}\,y(x)\right)$$

> **cond:=y(0)=500;**

$$cond := y(0) = 500$$

> **dsolve({de,cond},y(x));**

$$y(x) = \frac{20000}{1 + 39\,\mathbf{e}^{\left(-\frac{3x}{5}\right)}}$$

> y(s):=%;

$$y(s) := y(x) = \frac{20000}{1 + 39\,\mathbf{e}^{\left(-\frac{3x}{5}\right)}}$$

> **evalf(subs(x=3,y(s)));**

$$y(3) = 2685.769060$$

Таким образом, через три года численность населения горнорудного поселка составит ≈ 2685 человек.

Контрольные вопросы.

1. Что такое команды прямого и отложенного исполнения? Опишите их действия.
2. С помощью какой команды вычисляются пределы? Какие у нее параметры?
3. Какие команды позволяют найти производную функции?
4. Опишите команды, позволяющие исследовать функцию на непрерывность.
5. Какая последовательность команд необходима для нахождения max и min функции с указанием их координат (x, y)?
6. Какие недостатки имеют команды **maximize**, **minimize** и **extrema**?
7. Какие команды производят аналитическое и численное интегрирование? Опишите их параметры. Для чего предназначен пакет **student**?
8. Опишите команду интегрирования по частям.
9. Опишите команду интегрирования методом замены переменных.
10. Какая команда позволяет решить дифференциальное уравнение? Опишите ее параметры.
11. С помощью каких операторов обозначается производная в дифференциальном уравнении и в начальных условиях?
12. Опишите, как в *Maple* вычисляются частные производные.
13. Какие команды используются для вычисления двойных и тройных интегралов? Опишите их параметры.

Глава V. Линейная алгебра в профессиональной деятельности экономиста

§1. Векторная алгебра

Основная часть команд для решения задач линейной алгебры содержится в библиотеке **linalg**. Поэтому перед решением задач с матрицами и векторами следует загрузить эту библиотеку командой **with(linalg)**.

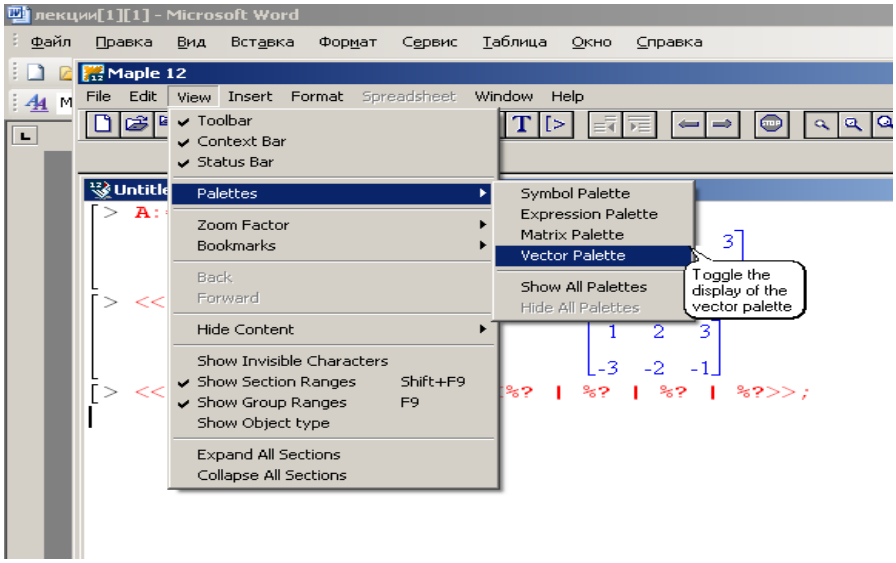
Способы задания векторов.

Для определения вектора в *Maple* используется команда **vector([x1,x2,…,xn]),** где в квадратных скобках через запятую указываются координаты вектора. *Например:* вектор выпуска продукции

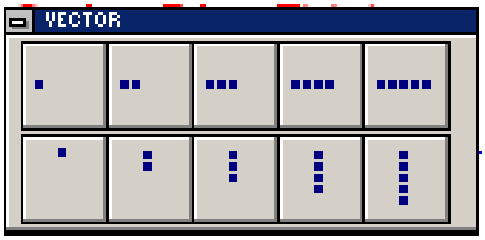
> **x:=vector([50,80,20,120]);**

$$x := [50, 80, 20, 120]$$

или при помощи меню View



Появляется диалоговое окно



и мы выбираем вектор необходимого размера, после чего на экране появляется запись

> **<%? | %? | %?>;**

где вместо знака %? записываем значения элементов вектора.

Координату уже определенного вектора **x** можно получить в строке вывода, если ввести команду **x[i]** , где **i** − номер координаты. *Например*, первую координату (количество продукции 1-го вида) заданного в предыдущем примере вектора выпуска продукции можно вывести так:

> **x[1];**

Вектор можно преобразовать в список и, наоборот, с помощью команды **convert(vector, list)** или **convert(list,vector).**

Сложение векторов.

Сложить два вектора ***a*** и ***b*** можно с помощью двух команд:

1) **evalm(a+b);**
2) **matadd(a,b).**

Команда **add** позволяет вычислять линейную комбинацию векторов ***a*** и ***b***: $\alpha\boldsymbol{a} + \beta\boldsymbol{b}$, где α,β − скалярные величины, если использовать формат: matadd(a,b,alpha,beta).

Скалярное, векторное произведение векторов и угол между векторами.

Скалярное произведение двух векторов $(a,b)=\sum_{i=1}^{n} a_i b_i$ вычисляется командой **dotprod(a,b).**

Векторное произведение двух векторов [***a***, ***b***] вычисляется командой **crossprod(a,b).**

Угол между двумя векторами ***a*** и ***b*** вычисляется с помощьюкоманды **angle(a,b)**.

Пример: определить суммарный расход сырья S (скалярное произведение вектора выпуска продукции и вектора расхода сырья), если предприятие выпускает 4 вида продукции P_1, P_2, P_3, P_4 в количествах 50, 80, 20,120 единиц, а нормы расхода сырья составляют соответственно 7; 3,5; 10; 4 кг.

```
> with(linalg):
> x:=vector([50,80,20,120]);
> y:= vector([7,3.5,10,4]);
> dotprod(x,y);
```

$$x := [\,50, 80, 20, 120\,]$$
$$y := [\,7, 3.5, 10, 4\,]$$
$$1310.0$$

Таким образом, суммарный расход сырья составит 1310 кг.

Норма вектора.

Норму (длину) вектора ***a*** = (x_1,..., x_n) , которая равна $\|a\|=\sqrt{x_1^2+x_2^2+...+x_n^2}$ ***a*** = *x*1 + ... + *x n* , можно вычислить с помощью команды **norm(a,2)**.

Можно нормировать вектор ***a*** с помощью команды **normalize(a),** в результате выполнения которой будет получен вектор единичной длины $\frac{a}{\|a\|}$.

Нахождение базиса системы векторов.

Если имеется система n векторов {a1, a2,...,an} , то с помощью команды basis([a1,a2,…,an]) можно найти базис этой системы.

§2. Действия с матрицами

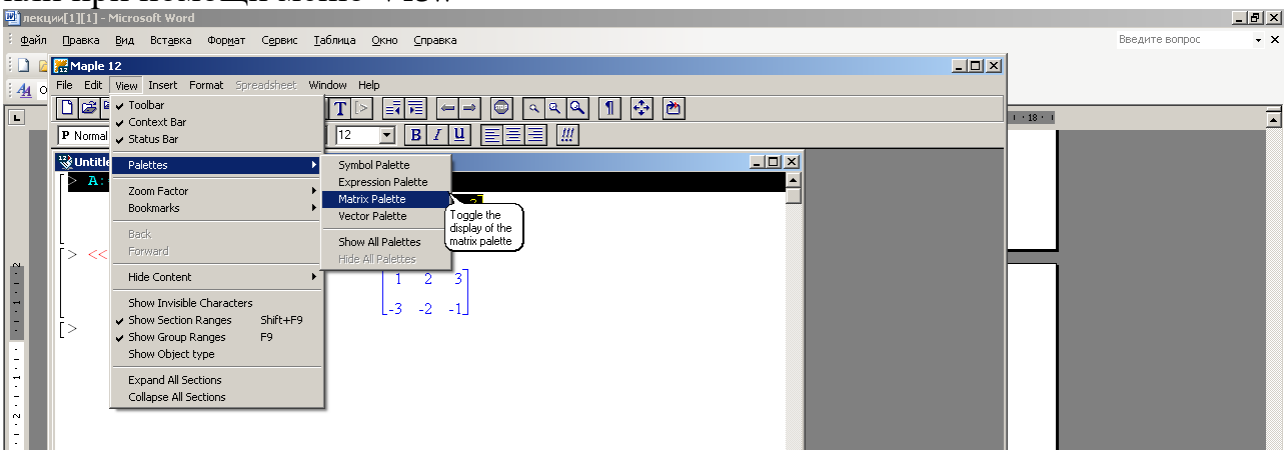
Для определения матрицы в *Maple* можно использовать команду **matrix(n, m, [[a11,a12,…,a1n], [a21,a22,…,a2m],…,[an1,an2,…,anm]]),** где **n** − число строк, **m** – число столбцов в матрице. Эти числа задавать необязательно, а достаточно перечислить элементы матрицы построчно в квадратных скобках через запятую.

Например: обувная фабрика выпускает продукцию трех видов (сапоги, туфли и кроссовки) и поставляет ее ежемесячно в течение зимы в города *A* и *B,* тогда *Z* – матрица зимних поставок

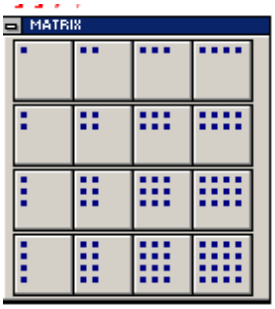
```
> Z:=matrix([[150,50,50],[200,70,60]]);
```

$$Z := \begin{bmatrix} 150 & 50 & 50 \\ 200 & 70 & 60 \end{bmatrix}$$

или при помощи меню View



Появляется диалоговое окно



и мы выбираем матрицу необходимого размера, после чего на экране появляется запись

> **<<%? | %? | %? | %?> , <%? | %? | %? | %?>>;**

где вместо %? записываем значения элементов матрицы.

В *Maple* матрицы специального вида можно генерировать с помощью дополнительных команд. В частности диагональную матрицу можно получить командой **diag**. *Например*: на приусадебном участке семья выращивает три сорта цветов с использованием трех видов удобрений. Нормы расхода удобрений заданы как элементы матрицы *A*.

Вид удобрения

$$A=\begin{pmatrix}4&0&0\\0&1&0\\0&0&3\end{pmatrix}$$ Сорт цветов

> **A:=diag(4,1,3);**

$$A:=\begin{bmatrix}4&0&0\\0&1&0\\0&0&3\end{bmatrix}.$$

Генерировать матрицу можно с помощью функции $f(i, j)$ от переменных i, j – индексов матрицы: **matrix(n, m, f),** где где **n** -число строк, **m** – число столбцов. Например:

> **f:=(i, j)->x^i*y^j;**

$$f := (i, j) \to x^i y^j$$

> **A:=matrix(2,3,f);**

$$A := \begin{bmatrix} x\,y & x\,y^2 & x\,y^3 \\ x^2\,y & x^2\,y^2 & x^2\,y^3 \end{bmatrix}$$

Число строк в матрице A можно определить с помощью команды **rowdim(A)**, а число столбцов – с помощью команды **coldim(A)**.

Арифметические операции с матрицами.

Сложение двух матриц одинаковой размерности осуществляется теми же командами, что и сложение векторов: **evalm(A+B)** или **matadd(A,B).**

Произведение двух матриц может быть найдено с помощью двух команд:

1) **evalm(A&*B);**

2) **multiply(A,B).**

В качестве второго аргумента в командах, вычисляющих произведение, можно указывать вектор, *например*: в некоторой отрасли m заводов выпускают n видов продукции, матрица А задает объемы продукции на каждом заводе в первом квартале, матрица В – соответственно во втором. Найти: а) объемы продукции; б) прирост объемов производства во втором квартале по сравнению с первым по видам продукции и заводам; в) стоимостное выражение выпущенной продукции за полгода (в долларах), если λ – курс доллара по отношению к рублю ($\lambda = (31, 31.5, 31.9, 32.4)$).

> A**:=matrix([[2,3,7],[1,2,2], [4,1,5],[2,1,3]]);**

> B:=matrix([[3,0,2],[2,4,1], [4,3,2],[5,2,4]]);

$$A := \begin{bmatrix} 2 & 3 & 7 \\ 1 & 2 & 2 \\ 4 & 1 & 5 \\ 2 & 1 & 3 \end{bmatrix}$$

$$B := \begin{bmatrix} 3 & 0 & 2 \\ 2 & 4 & 1 \\ 4 & 3 & 2 \\ 5 & 2 & 4 \end{bmatrix}$$

а) объемы продукции за полугодие определяются суммой матриц

> **V:=matadd(A,B);**

$$V := \begin{bmatrix} 5 & 3 & 9 \\ 3 & 6 & 3 \\ 8 & 4 & 7 \\ 7 & 3 & 7 \end{bmatrix}$$

б) прирост во втором квартале по сравнению с первым определяется разностью матриц:

> **P:=evalm(B-A);**

$$P := \begin{bmatrix} 1 & -3 & -5 \\ 1 & 2 & -1 \\ 0 & 2 & -3 \\ 3 & 1 & 1 \end{bmatrix}$$

Отрицательные элементы p_{ij} показывают, что на данном заводе i объем производства j-го продукта уменьшился; положительные p_{ij} – увеличился; нулевые p_{ij} – не изменился.

в) Произведение $\lambda V = \lambda(A+B)$ дает выражение стоимости объемов производства за квартал в долларах по каждому заводу и каждому предприятию:

> **lambda:=vector([31,31.5,31.9,32.4]);**

$$\lambda := [31, 31.5, 31.9, 32.4]$$

> **multiply(lambda,V);**

$$[731.5, 506.8, 823.6]$$

Команда **evalm** позволяет также прибавлять к матрице число и умножать матрицу на число. *Например*:

> **C:=matrix([[1,1],[2,3]]):**

> **evalm(2+3*C);**

$$\begin{bmatrix} 5 & 3 \\ 6 & 11 \end{bmatrix}$$

Определители, миноры и алгебраические дополнения. Ранг и след матрицы.

Определитель матрицы A вычисляется командой **det(A).** Команда **minor(A,i,j)** возвращает матрицу, полученную из исходной матрицы A вычеркиванием i-ой строки и j-ого столбца.

Минор Mij элемента aij матрицы A можно вычислить командой d**et(minor(A,i,j)).** Ранг матрицы A вычисляется командой **rank(A).** След матрицы A, равный сумме ее диагональных элементов, вычисляется командой **trace(A).**

> **A:=matrix([[4,0,5],[0,1,-6],[3,0,4]]);**

$$A := \begin{bmatrix} 4 & 0 & 5 \\ 0 & 1 & -6 \\ 3 & 0 & 4 \end{bmatrix}$$

> **det(A);**

$$1$$

> **trace(A);**

$$9 .$$

Обратная и транспонированная матрицы.

Обратную матрицу A^{-1} , такую что $A^{-1}A=A\,A^{-1}=E$, где E − единичная матрица, можно вычислить двумя способами:

1) **evalm(1/A);**

2) **inverse(A).**

Транспонирование матрицы *A* – это изменение местами строк и столбцов. Полученная в результате этого матрица называется транспонированной и обозначается *A′*. Транспонированную матрицу *A′* можно вычислить командой **transpose(A).**

Например, используя заданную в предыдущем пункте матрицу *A*, найдем ей обратную и транспонированную:

> **evalm(1/A);**

$$\begin{bmatrix} 4 & 0 & -5 \\ -18 & 1 & 24 \\ -3 & 0 & 4 \end{bmatrix}$$

> **inverse(A);**

$$\begin{bmatrix} 4 & 0 & -5 \\ -18 & 1 & 24 \\ -3 & 0 & 4 \end{bmatrix}$$

> **transpose(A);**

$$\begin{bmatrix} 4 & 0 & 3 \\ 0 & 1 & 0 \\ 5 & -6 & 4 \end{bmatrix}.$$

Функции от матриц.

Возведение матрицы *A* в степень *n* производится командой **evalm(A^n).**

Собственные числа и собственные векторы матрицы.

Из курса линейной алгебры известно, что если Ax=λx, то вектор x называется собственным вектором матрицы A, а число λ – вектору. Совокупность всех собственных чисел матрицы называется спектром матрицы. Если в спектре матрицы одно и тоже собственное число встречается k раз, то говорят, что кратность этого собственного числа равна k.

Для нахождения собственных чисел матрицы A используется команда eigenvalues(A). Для нахождения собственных векторов матрицы A используется команда eigenvectors(A). В результате выполнения этой команды будут получены собственные числа, их кратность и соответствующие собственные векторы.

Чтобы понять, в каком виде получаются результаты выполнения команды **eigenvectors,** внимательно разберитесь со следующим примером:

Матрица $A := \begin{bmatrix} 0.2 & 0.3 & 0.2 \\ 0.6 & 0.4 & 0.6 \\ 0.2 & 0.3 & 0.2 \end{bmatrix}$ имеет 3 собственных вектора:

a1 = (0.722998,-0.4$10^{-9}$,-0.722998), отвечающий собственному числу λ1 = $-0.1 \cdot 10^{-9}$ кратности 1,

a2 = (0.442325,0.884651, 0.442325) , отвечающий собственному числу λ2 = 1 кратности 1,

a3 = (-0.390867,0.781735, -0.390867) , отвечающий собственному числу λ3 = -0.19999 кратности 1.

> **A:=matrix([[0.2,0.3,0.2],[0.6,0.4,0.6],[0.2,0.3,0.2]]);**

$$A := \begin{bmatrix} 0.2 & 0.3 & 0.2 \\ 0.6 & 0.4 & 0.6 \\ 0.2 & 0.3 & 0.2 \end{bmatrix}$$

> **with(linalg):eigenvectors(A);**

$[-0.1\ 10^{-9}, 1, \{[0.7229988056, -0.4\ 10^{-9}, -0.7229988059]\}],$
$[1.000000000, 1, \{[0.4423258680, 0.8846517375, 0.4423258695]\}],$
$[-0.1999999999, 1, \{[-0.3908679804, 0.7817359607, -0.3908679791]\}]$

В строке вывода перечислены в квадратных скобках собственное число, его кратность и соответствующий собственный вектор в фигурных скобках, затем следующие наборы таких же данных. Полученный результат означает, что сбалансированность торговли трех стран достигается при отношении их национальных доходов $1:2:1$, (отвечающие собственному значению $\lambda = 1$).

Характеристический и минимальный многочлены матрицы.

Для вычисления характеристического многочлена $P_A(\lambda) = \det(\lambda E - A)$ матрицы A используется команда **charpoly(A,lambda).**

Минимальный многочлен (делитель) матрицы A можно найти с помощью команды **minpoly(A,lambda).**

Канонические и специальные виды матрицы.

К треугольному виду матрицу A можно привести тремя способами:

1) команда gausselim(A) приводит матрицу A к треугольному виду методом Гаусса;

2) команда ffgausselim(A) приводит матрицу A к треугольному виду методом Гаусса без деления. Эта команда предпочтительней для работы с символьными матрицами, так как не производит нормировку элементов и исключает возможные ошибки, связанные с делением на нуль;

3) команда gaussjord(A) приводит матрицу A к треугольному виду методом Гаусса-Жордана.

Характеристическую матрицу $F(A) = \lambda E - A$ можно вычислить командой charmat(A,lambda).

§3. Системы линейных уравнений. Матричные уравнения

Система линейных уравнений $Ax = b$ может быть решена двумя способами.

Способ 1: стандартная команда solve находит решение системы линейных уравнений, записанных в развернутом виде.

Способ 2: команда linsolve(A,b) из пакета linalg находит решение уравнения $Ax = b$. Аргументы этой команды: A – матрица, b – вектор.

С помощью команды linsolve(A,b) можно найти решение матричного уравнения AX=B, если в качестве аргументов этой команды указать, соответственно, матрицы A и B.

Пример. Обувная фабрика специализируется по выпуску изделий трех видов: сапог, кроссовок и ботинок; при этом используется сырье трех типов: S_1, S_2, S_3. Нормы расхода каждого из них на одну пару обуви и объем расхода сырья на 1 день заданы таблицей:

Вид сырья	Нормы расхода сырья на одну пару, усл.ед.			Расход сырья на 1 день, усл.ед.
	Сапоги	Кроссовки	Ботинки	
S_1	5	3	4	2700
S_2	2	1	1	900
S_3	3	2	2	1600

Найти ежедневный объем выпуска каждого вида обуви.

Решение: Пусть ежедневно фабрика выпускает x_1 пар сапог, x_2 пар кроссовок и x_3 пар ботинок. Тогда в соответствии с расходом сырья каждого вида имеем систему:

$$\begin{cases} 5x_1 + 3x_2 + 4x_3 = 2700 \\ 2x_1 + x_2 + x_3 = 900 \\ 3x_1 + 2x_2 + 2x_3 = 1600 \end{cases}$$

> **eq:={5*x[1]+3*x[2]+4*x[3]=2700, 2*x[1]+x[2]+x[3]=900, 3*x[1]+2*x[2]+2*x[3]=1600}:**

> **s:=solve(eq,{x[1],x[2],x[3]});**

$$s := \{x_1 = 200, x_2 = 300, x_3 = 200\}.$$

Таким образом, фабрика выпускает: 200 пар сапог, 300 – кроссовок и 200 пар ботинок.

Контрольные вопросы.

1. Какой пакет следует загрузить перед решением задач линейной алгебры в *Maple*?
2. С помощью каких команд можно ввести вектор, матрицу?
3. Какими двумя командами можно сложить два вектора одинаковой размерности (2 матрицы)?
4. Какие виды произведений векторов вычисляются *Maple* и какие команды для этого используются?
5. Как вычислить норму вектора?
6. Опишите команды нахождения базиса системы векторов.
7. Какие команды используются для нахождения определителя, минора, алгебраического дополнения, следа матрицы?

8. Какая матрица называется обратной и какими способами она вычисляется в *Maple*?
9. Что называется собственным вектором и собственным числом матрицы? Что называется спектром матрицы? Какие команды используются для нахождения спектра матрицы и ее собственных векторов? В каком виде в *Maple* выводятся результаты выполнения этих команд?
10. Перечислите специальные виды матриц и команды, приводящие матрицы к этим формам.
11. Какая команда позволяет решать матричные уравнения?

Глава VI. МАТЕМАТИЧЕСКИЕ МОДЕЛИ В ЭКОНОМИКЕ

§ 1. Постановка задачи. Метод наименьших квадратов

С учетом неизбежных погрешностей исходных данных при подборе заменяющей их кривой можно снять требование обязательного прохождения этой кривой через заданные точки, заменив его требованием достаточной близости кривой к точкам.

Тогда задачу можно сформулировать следующим образом: для «облака» точек x_i , y_i (i =1, 2, ...n) подобрать кривую $y(x)$, которая давала бы значения $y(x_i)$, достаточно близкие к y_i.

Кривая $y(x)$ называется аппроксимирующей кривой или линией регрессии. Для единственности решения нужно, как и при интерполяции, оговорить класс аппроксимирующей кривой и, кроме того, критерий ее близости к точкам исходных данных. В методе наименьших квадратов таким критерием является минимум суммы квадратов отклонений ординат $y(x_i)$ линии регрессии от ординат y_i экспериментальных точек. Эта сумма имеет вид:

$$S = \sum_{i=1}^{n} (y_i - y(x_i))^2. \qquad (1)$$

В качестве класса аппроксимирующей функции часто выбираются степенные полиномы

$$y(x) = a_0 + a_1 x + a_2 x^2 + ... + a_m x^m \text{ ; } (m < n). \qquad (2)$$

Линейная аппроксимация.

Рассмотрим наиболее простой случай, когда m = 1. Аппроксимирующая функция является прямой линией $y(x) = a_0 + a_1 x$. Уравнение (1) принимает вид:

$$S = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2. \qquad (3)$$

Для минимизации S вычислим частные производные от S по a_o и a_1 и приравняем их нулю. Получим следующую систему линейных уравнений:

$$\sum_{i=1}^{n} (y_i - a_0 - a_1 x_i) = 0;$$

$$\sum_{i=1}^{n}(y_i - a_0 - a_1 x)(-x_i) = 0;$$

решая которую относительно неизвестных коэффициентов, найдем:

$$a_1 = \frac{\sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i - n\sum_{i=1}^{n} x_i\, y_i}{\left(\sum_{i=1}^{n} x_i\right)^2 - n\sum_{i=1}^{n} x_i^2}; \qquad (4)$$

$$a_0 = 1/n \left(\sum_{i=1}^{n} y_i - a_1 \sum_{i=1}^{n} x_i\right).$$

Другие виды аппроксимации.

С увеличением степени полинома (2) схема определения его коэффициентов остается прежней, однако число уравнений увеличивается и расчеты усложняются. Приходится решать систему *m+1* уравнений.

В более общем случае искомые величины входят в участвующие зависимости нелинейно, метод приводит к системе нелинейных уравнений, и вычислительные трудности особенно возрастают. Отметим, что в некоторых частных случаях нелинейную аппроксимацию можно привести к линейной заменой переменных или логарифмированием.

Использование Maple

1 способ. Рассмотрим пример Maple для подбора регрессий. Соответствующая функция LeastSquares находится в пакете **CurveFitting**.

Подбор регрессий

Используем для примера следующие данные экспериментов по измерениям зависимой переменной *y* в функции независимой *x*, заданные в виде списков:

```
> x:=[1,2,3,4,5,6];y:=[1,4,5.5,4.5,2.8,6.5];
```

$$x := [1, 2, 3, 4, 5, 6]$$

$$y := [1, 4, 5.5, 4.5, 2.8, 6.5]$$

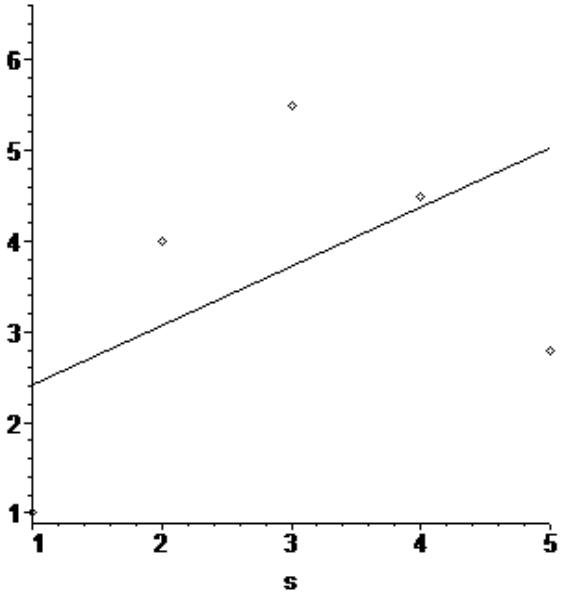
```
> with(CurveFitting):
> r:=LeastSquares([[1,1],[2,4],[3,5.5],[4,4.5],[5,2.8],[6,6.5]],s);
```

$$r := 1.760000000 + 0.6542857143\ s$$

```
> gr1:=plot(r,s=1..5,color=black):
> L:=zip((x,y)->[x,y],(X,Y));
```

$$L := [[1, 1], [2, 4], [3, 5.5], [4, 4.5], [5, 2.8], [6, 6.5]]$$

```
> with(plots):gr2:=pointplot(L,color=black):
> display(gr1,gr2);
```



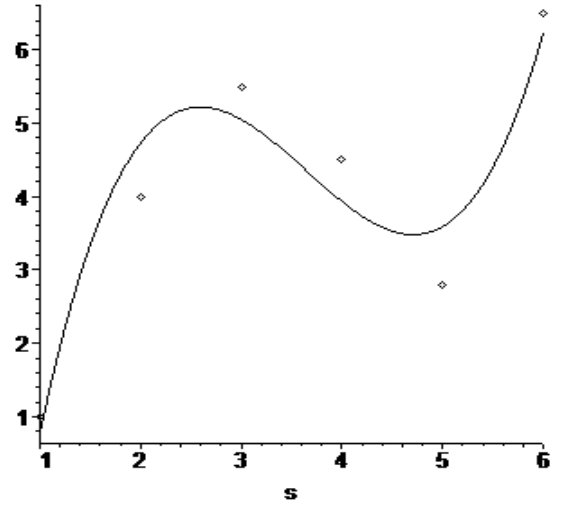
Как видно из построенного графика, линейная регрессия (прямая линия) плохо аппроксимирует экспериментальные точки. Подберём кубическую регрессию (полином 3-й степени).

> **r1:=LeastSquares(X,Y,s,curve=a*s^3+b*s^2+c*s+d);**

$$r1 := -9.100000000 + 13.52817460\ s - 4.045238095\ s^2 + 0.3694444444\ s^3$$

> **gr3:=plot(r1,s=1..6,color=black):**

> **display(gr2,gr3);**



Кубический полином соответствует экспериментальным точкам более точно. Отметим, что используемая функция позволяет использовать регрессию в виде полинома со старшей степенью до *n-1*, где *n* – число экспериментальных точек.

2 способ.

Регрессионный анализ.

Для проведения регрессионного анализа служит функция fit, которая вызывается следующим образом:

> stats[fit,leastsquare[vars,eqn.parms]](data);

или

> fit[leastsquare[vars,eqn.parms]](data);

где data — список данных, vars — список переменных для представления данных, eqn — уравнение, задающее аппроксимирующую зависимость (по умолчанию линейную), parms — множество параметров, которые будут заменены вычисленными значениями.

Пример.

> with(stats):

> Digits:=5;

$$Digits := 5$$

> fit[leastsquare[[x,y]]]([[1,2,3,4],[3,3.5,3.9,4.6]]);

$$y = 2.4500 + .52000\,x$$

> fit[leastsquare[[x,y],y=a*x^2+b*x+c]]([[1,2,3,4], [3,3.5,3.9,4.6]]);

$$y = .050000\,x + .27000\,x + 2.7000$$

В этом примере функция регрессии не задана, поэтому реализуется простейшая линейная регрессия, и функция fit возвращает полученное уравнение регрессии для исходных данных, представленных списками координат узловых точек. Это уравнение аппроксимирует данные с наименьшей среднеквадратичной погрешностью.

Функция fit может обеспечивать регрессию и для функций нескольких переменных. При этом надо просто увеличить размерность массивов исходных данных.

Пример.

> f:=fit[leastsquare[[x1,x2,y],y=-
a+b*x1+c*x2,{a,b,c}]]\([[1,2,3.5,5],[2,4,6,8.8],[3,5,7,10]]);

$$f := y = .86066 - .16393\,x1 + 1.1270\,x2$$

> fa:=unapply(rhs(f),x1,x2);

$$fa := (x1,x2) \rightarrow .86066 - .16393\,x1 + 1.1270\,x2$$

> fa(1,2);

2.9507

В данном случае уравнение регрессии задано в виде $y = -a + bx_1 + cx_2$.

Замечание. Применение полученной функции регрессии для вычислений или построения ее графика. Прямое применение функции f в данном случае невозможно, так как она представлена в не вычисляемом формате. Для получения вычисляемого выражения она преобразуется в функцию двух переменных fa(x1,x2) путем отделения правой части выражения для функции f. После этого возможно вычисление значений функции fa(x1,x2) для любых заданных значений x1 и x2:

Функция fit неприменима для нелинейной регрессии. При попытке ее проведения возвращается структура процедуры, но не результат регрессии.

§ 2. Линейное программирование

Пакет simplex содержит команды для решения задач линейного программирования, при помощи симплекс-метода. Перед обращением к командам пакета нужно его подгрузить. Для нахождения максимума функции используется команда **maximize,** для нахождения минимума функции **minimize.**

1 способ.

Пример. Решить задачу линейного программирования с помощью пакета simplex.

```
with(simplex):
f:=0.5*x[1]+2*x[2];
u1:=x[1]+x[2]<=6;
u2:=x[1]-x[2]<=1;
u3:=0.5*x[1]-x[2]>=-4;
u4:=2*x[1]+x[2]>=6;
u5:=x[1]>=1;
u6:=x[2]>=1;
maximize(f,{u1,u2,u3,u4,u5,u6},NONNEGATIVE);
```

$$f := 0.5\,x_1 + 2\,x_2$$

$$u1 := x_1 + x_2 \le 6$$

$$u2 := x_1 - x_2 \le 1$$

$$u3 := -4 \le 0.5\,x_1 - x_2$$

$$u4 := 6 \le 2\,x_1 + x_2$$

$$u5 := 1 \le x_1$$

$$u6 := 1 \le x_2$$

$$\{x_1 = 1.333333333\,,\ x_2 = 4.666666667\ \}$$

Пример. Решить задачу линейного программирования с помощью пакета simplex.

```
f:=x[1]+x[2]+x[3]+x[4];
u1:=x[1]+x[2]>=0;
u2:=x[1]+x[2]-x[3]+x[4]>=1;
u4:=x[1]>=1;
u5:=x[1]<=2;
u3:=x[2]+x[3]>=1;
minimize(f,{u1,u2,u3,u4,u5},NONNEGATIVE);
```

$$f := x_1 + x_2 + x_3 + x_4$$

$$u1 := 0 \le x_1 + x_2$$

$$u2 := 1 \le x_1 + x_2 - x_3 + x_4$$

$$u4 := 1 \le x_1$$

$$u5 := x_1 \le 2$$

$$u3 := 1 \le x_2 + x_3$$

$$\{x_1 = 1, x_2 = 1, x_3 = 0, x_4 = 0\}$$

2 способ.

Пример.

> **with(simplex):F:=-x+2*y+3*z;**

$$F := -x + 2y + 3z$$

> **cns:={x+2*y-3*z<=4,5*x-6*y+7*z<=8,9*x+10*z<=11}:**

> **maximize(F,cns union{x>=0,y>=0});**

$$\{x = 0, y = \frac{73}{20}, z = \frac{11}{10}\}$$

.

Контрольные вопросы.

1. Какой пакет следует загрузить для подбора регрессий?
2. Какой пакет следует загрузить для решения задач линейного программирования в *Maple*?
3. Для чего в *Maple* предназначена функция fit?

Литература

1. Высшая математика для экономистов: учебник для вузов/Н.Ш. Кремер, Б,А. Путко, И.М. Тришин, М.Н. Фридман; Под ред.проф. Н.Ш. Кремера. – М.: ЮНИТИ, 2004. – 471с.
2. Дьяконов В. Maple 7: учебный курс. – СПб.: Питер, 2002.– 669с.
3. Замков О.О., Толстопятенко А.В., Черемных Ю.Н. Математические методы в экономике: учебник/Под.общ.ред.д.э.н., проф. А.В. Сидоровича – М.: Издательство «Дело и Сервис», 2001. – 368с.
4. Красс М.С., Чупрынов Б.П. Основы математики и ее приложения в экономическом образовании: Учебник. – М.: 2008. – 719с.
5. Савотченко С.Е., Кузьмичева Т.Г. Методы решения математических задач в Maple: Учебное пособие – Белгород: Изд. Белаудит, 2001. – 116 с.
6. Сборник задач по высшей математике для экономистов /Геворкян П.С. и др.; Под ред. П.С. Геворкяна. — М.: ЗАО. Издательство. Экономика., 2010. — 384 с.
7. Сдвижников О.А. Математика на компьютере: Maple 8. – М.: СОЛОН – Пресс, 2003. – 176с.

Лебедева Е.В.

ИСПОЛЬЗОВАНИЕ ИНТЕГРИРОВАННЫХ СИМВОЛЬНЫХ СИСТЕМ ПРИ РЕШЕНИИ ЭКОНОМИКО-МАТЕМАТИЧЕСКИХ ЗАДАЧ

Учебно-методическое пособие

Подписано в печать 23.11.2010. Формат 60x80 1/16
Печать оперативная. Бумага офсетная. Гарнитура Times.
Объем 2,5 усл. п.л. Тираж 500 экз. Заказ №781

Отпечатано с готового оригинала макета
на полиграфической базе
редакционно-издательского отдела
ФГБОУ ВПО «Орловский государственный университет»,
302026 г. Орел, ул. Комсомольская, 95
Тел. (4862) 74-45-08