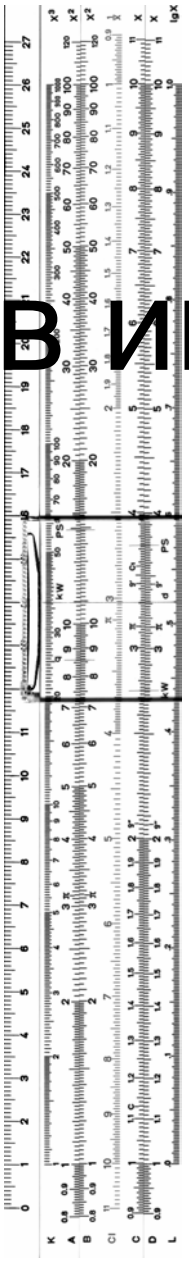
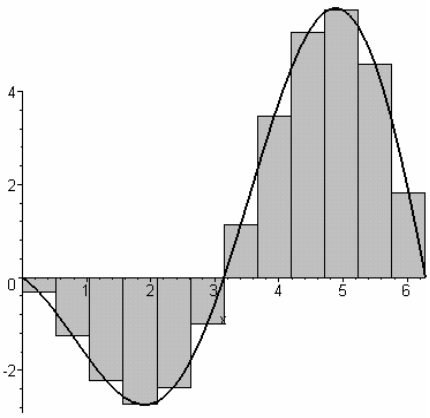
А. А. Коптев
А. А. Пасько
А. А. Баранов



maple в инженерных расчетах



Издательство ТГТУ

Министерство образования Российской Федерации

Тамбовский государственный технический университет

А. А. Коптев, А. А. Пасько, А. А. Баранов

Maple в инженерных расчетах

Учебное пособие

Утверждено Ученым советом университета
в качестве учебного пособия

Тамбов
Издательство ТГТУ
2003

УДК 519.67
ББК з973-018.2
К55

Рецензенты:
Кандидат технических наук, доцент
В. А. Богуш
Доктор технических наук, профессор
С. И. Дворецкий

Коптев А. А., Пасько А. А., Баранов А. А.
К55 Maple в инженерных расчетах: Учеб. пособие. Тамбов: Изд-во Тамб. гос. техн. ун-та, 2003. 80 с.
ISBN 5-8265-0211-8

Учебное пособие посвящено системе символьных вычислений *Maple*. Представлены основные понятия языка *Maple* и наиболее часто используемые функции. Подробно рассмотрены вопросы графического отображения полученных с помощью *Maple* решений. Дано введение в программирование на языке *Maple*. Особое внимание уделено применению *Maple* для решения задач расчета машин и аппаратов химических производств.

Пособие предназначено для студентов и аспирантов, использующих персональный компьютер для решения задач математического и прикладного характера.

УДК 519.67
ББК з973-018.2

ISBN 5-8265-0211-8

Учебное издание

КОПТЕВ Андрей Алексеевич,
ПАСЬКО Александр Анатольевич,
БАРАНОВ Андрей Алексеевич

Maple в инженерных расчетах

Учебное пособие

Редактор В. Н. Митрофанова
Компьютерное макетирование Е. В. Кораблевой

Подписано к печати 24.01.2003
Формат 60 × 84/16. Бумага офсетная. Печать офсетная
Объем: 4,65 усл. печ. л.; 4,56 уч. изд. л.
Тираж 200 экз. С. 40

Издательско-полиграфический центр ТГТУ
392000, ТАМБОВ, СОВЕТСКАЯ, 106, К. 14

ВВЕДЕНИЕ

Вследствие непрерывно возрастающих требований к качеству, экономичности, надежности, быстродействию, снижению материалоемкости оборудования, современные инженерные расчеты все более усложняются. Они должны учитывать все факторы, оказывающие совокупное влияние на работу современных машин и аппаратов: режимы эксплуатации, свойства материалов, условия нагружения и т.п. При этом необходимо соблюдение основных показателей надежности, прочности и долговечности.

Поэтому в расчетах все шире применяются теоретические результаты, строго математически описывающие задачу, и все меньше используются ориентировочные, приближенные зависимости.

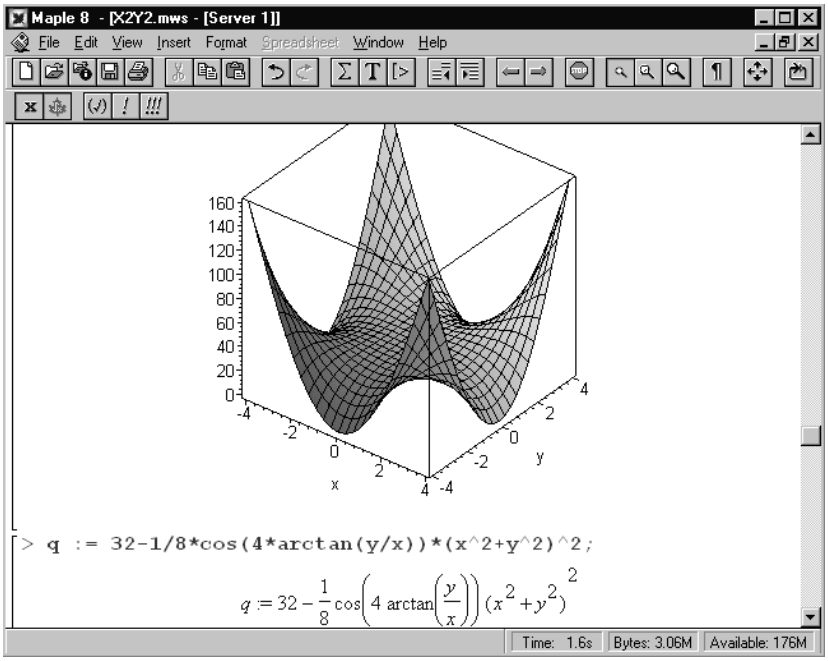
Решение математических задач, возникающих перед инженером, невозможно без умелого применения вычислительной техники и ее программного обеспечения. Существенно облегчить расчеты в инженерных задачах, повысить их качество и быстроту может универсальный математический пакет *Maple* компании *Waterloo Maple*, который по праву считается одной из лучших программ для выполнения, в первую очередь, аналитических математических расчетов. Ценность чисто аналитических конструкций, с которыми манипулирует *Maple*, позволяет избежать погрешностей неизбежных при численных решениях, получить удобные расчетные зависимости, увеличить производительность численных экспериментов. Однако в некоторых случаях без привлечения численных методов невозможно получить решение многих современных инженерных задач. В данной ситуации с успехом можно использовать универсальные численные алгоритмы, входящие в систему *Maple* с возможностью регулировать точность вычислений до порядков не доступных максимальным аппаратным значениям современных компьютеров.

В данном пособии описаны некоторые возможности пакета *Maple 8*, но многие из этих возможностей имеются и в более ранних версиях, при этом пробную версию пакета *Maple 8* можно получить на сервере компании *Waterloo Maple http://www.maplesoft.com*.

1 ИНТЕРФЕЙС *MAPLE*

Как любое *Windows* – приложение *Maple* имеет оконный интерфейс, строки команд которого будут несколько отличаться в зависимости от следующих действий:

- редактирование рабочего документа – стандартный интерфейс;
- рабочего листа;
- просмотр справки – интерфейс справочной системы;
- двухмерные построения – интерфейс графической двухмерной системы;
- трехмерные построения – интерфейс трехмерной графической системы.



1.1 ИНТЕРФЕЙС РАБОЧЕГО ДОКУМЕНТА

Интерфейс рабочего документа будет показан на экране, если пользователь работает в рабочем документе и там же расположен курсор ввода. Вид данного интерфейса был представлен на рисунке. Как видно из рисунка, в строке команд содержится восемь пунктов меню:

- ***File*** – команды для работы с файлами сессии *Maple*;
- ***Edit*** – команды для работы с отдельным регионом или его частью;
- ***View*** – изменение вида содержимого рабочего документа и панелей управления;
- ***Insert*** – вставка различных объектов и текста в открытый документ;
- ***Format*** – команды форматирования текста;
- ***Spreadsheet*** – команды для работы с электронными таблицами;
- ***Window*** – команды для закрытия, упорядочивания и вывода списка открытых рабочих документов;
- ***Help*** – команды для работы со справочной системой и изменения базы данных помощи.

Наиболее часто используемые команды управления рабочим документом вынесены в пиктографическое меню, описание которых приведено ниже.

 – открытие нового документа;

 – открытие существующего документа;

 – открытие URL;

 – сохраняет текущий документ в файле на диске;

 – печать активного документа;

 – вырезать выделенную часть документа и отправить его в буфер обмена;

 – копировать выделенную часть документа в буфер обмена;

 – вставить содержимое буфера обмена в активный документ;

 – отменить последнюю операцию редактирования;

 – вернуть отмененную операцию;

 – вставка команды *Maple* непосредственно в ту часть документа, где находится курсор;

 – вставка и форматирование тестового комментария;

 – вставка группы выполняемых команд;

 – преобразовать выделение в подсекцию;

 – действие, обратное предыдущему;

 – шаг назад при работе с гиперссылками;

 – шаг вперед при работе с гиперссылками;

 – прервать вычисления;

 масштаб отображения рабочего документа (100 %, 150 % и 200 % соответственно);

 – показать /скрыть специальные символы;

 – увеличить размер активного окна;

 – очистить внутреннюю память (restart);

 – переключает отображение строки команд из математической в *Maple*-нотацию и обратно;

 – выполнять/не выполнять выражение;

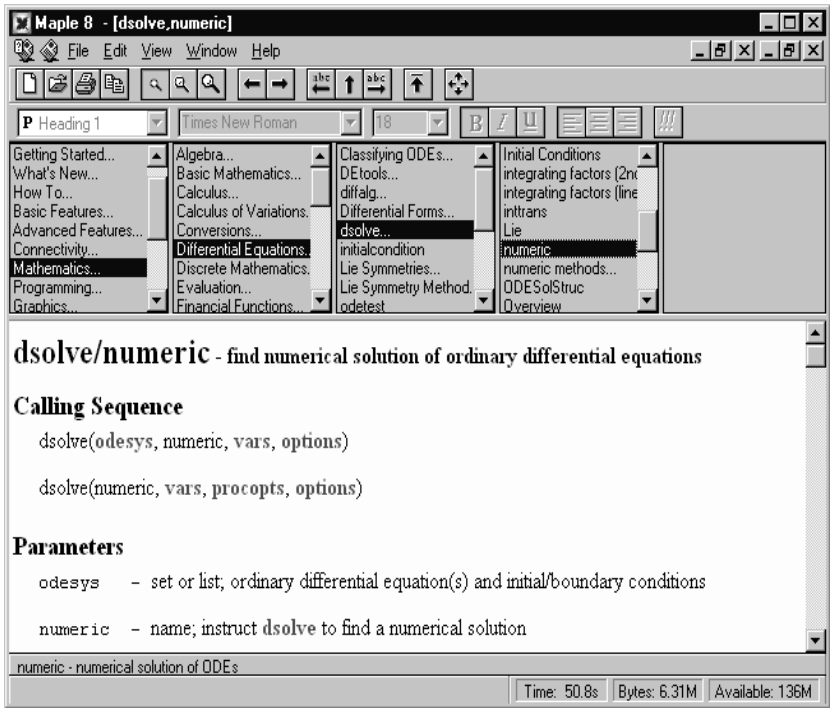
 – автоматическая коррекция синтаксиса выражения;

 – выполнить текущее выражение;

 – выполнить рабочий документ.

1.2 ИНТЕРФЕЙС СПРАВОЧНОЙ СИСТЕМЫ

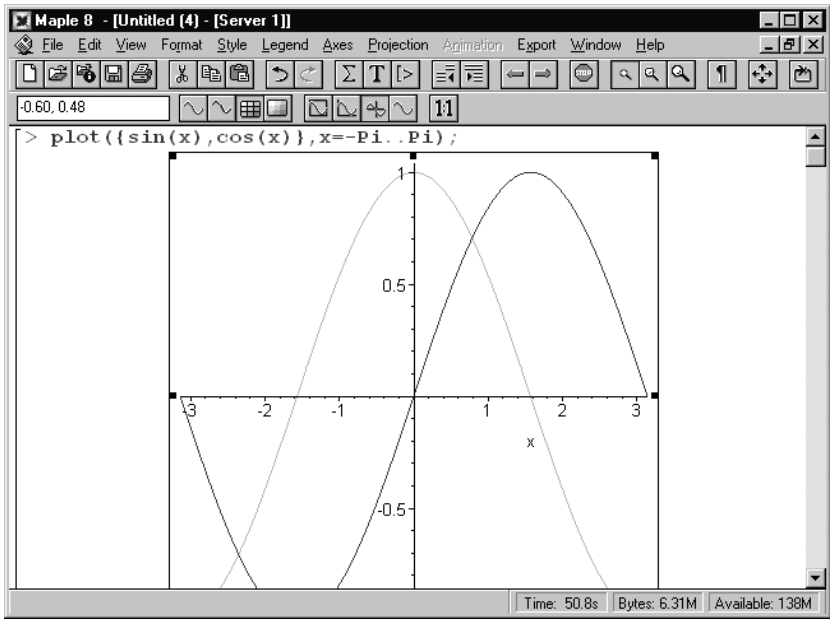
Maple снабжен мощной диалоговой системой контекстной настраиваемой помощи. При работе со справочной системой можно увидеть такую картину.



Справочную информацию можно искать по определенной теме или команде, а также по широкому диапазону доступных команд. Для получения справки по конкретной команде следует в рабочем документе ввести "?" и имя команды, либо установить курсор на интересующую команду и нажать клавишу **F1**.

1.3 ИНТЕРФЕЙС ДВУХМЕРНОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ

При выполнении графических построений на плоскости перед пользователем появляется интерфейс двухмерной графической сис- темы.

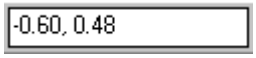


При этом командная строка содержит следующие пункты меню:

- ***File*** –стандартное меню интерфейса рабочего документа;

- ***Edit*** – стандартное меню интерфейса рабочего документа;
- ***View*** – стандартное меню интерфейса рабочего документа;
- ***Format*** – команды форматирования;
- ***Style*** – определяет стиль построения;
- ***Legend*** – редактирование и показ легенды;
- ***Axes*** – управляет стилем координатных осей;
- ***Projection*** – определяет масштаб изображения;
- ***Animation*** – анимация графиков;
- ***Export*** – сохранение графики в файлы различных форматов;
- ***Window*** – стандартное меню интерфейса рабочего документа;
- ***Help*** – стандартное меню интерфейса рабочего документа.

Наиболее часто используемые команды управления двухмерной графической системой вынесены в пиктографическое меню, описание которых приведено ниже.



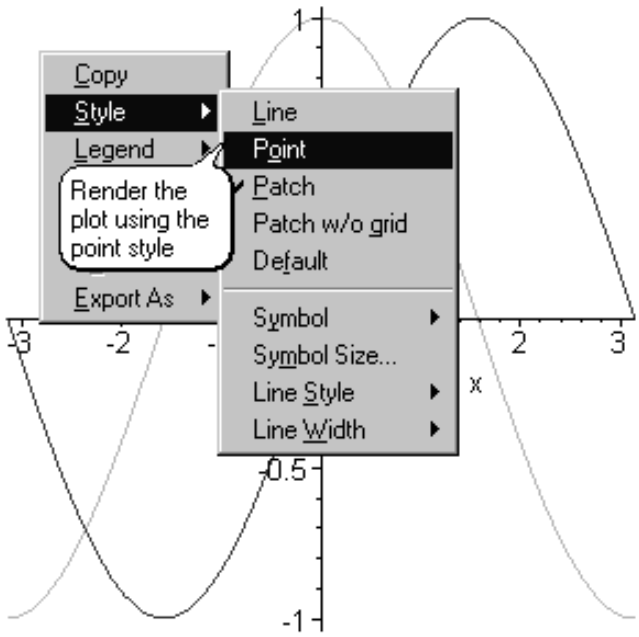
– координаты курсора;


– стили построения;


– стили координатных осей;

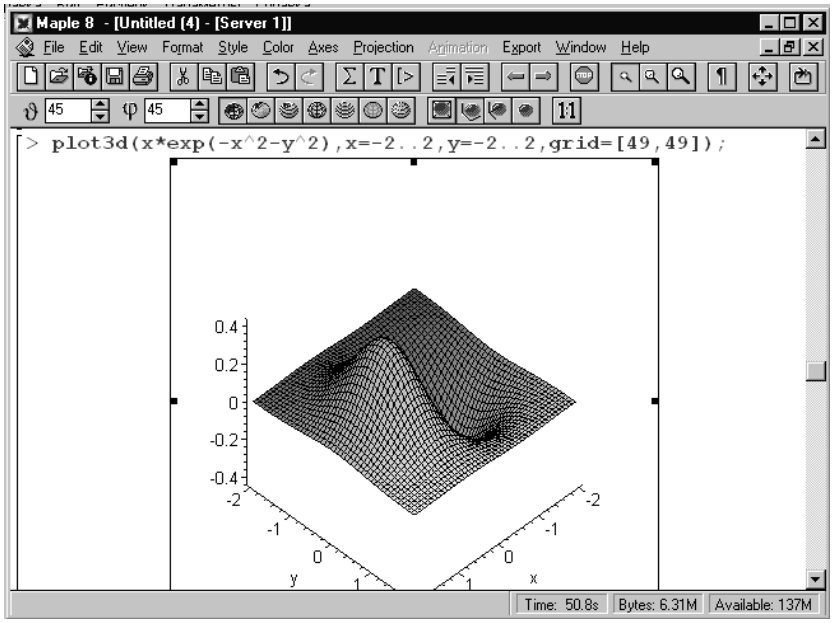
– масштабирование изображения.

Управлять двухмерной графической системой можно используя контекстное меню. Оно вызывается нажатием правой клавиши мыши на поле изображения.



1.4 ИНТЕРФЕЙС ТРЕХМЕРНОЙ ГРАФИЧЕСКОЙ СИСТЕМЫ

При любом виде трехмерного построения перед пользователем возникает интерфейс трехмерной графической системы.



Перечислим пункты меню, не описанные в стандартном интерфейсе рабочего документа:

- ***Style*** – определяет стиль построения;
- ***Color*** – определяет цвет построения;
- ***Axes*** – управляет стилем координатных осей;
- ***Projection*** – определяет масштаб изображения;
- ***Animation*** – анимация графиков;
- ***Export*** – сохранение графики в файлы различных форматов.

В пиктографическое меню вынесены следующие команды:

 – направление взгляда на объект;

 – стили построения;

 – стили координатных осей;

 – масштабирование изображения.

2 СИНТАКСИС ЯЗЫКА *MAPLE*

2.1 ПРОСТЫЕ ВЫЧИСЛЕНИЯ

В *Maple* выполняемые математические выражения вводятся всегда после символа **>**, а заканчиваются точкой с запятой или двоеточием, если результат не надо выводить на экран. Чтобы продолжить запись предложения на следующей строке используют комбинацию "Shift + Enter". При нажатии клавиши "Enter" предложение выполняется. Обнаружив ошибку, *Maple* выводит сообщение о ней в следующей строке.

> **1+2;**

$$3$$

> **12*4/3;**

$$16$$

> **1+3/2;**

$$\frac{5}{2}$$

> **1.125/2;**

0.5625000000

> **1/0;**

Error, numeric exception: division by zero

Целые числа в *Maple* имеют наивысший приоритет, поэтому $1+3/2=\frac{5}{2}$, а не 2.5. Всегда можно получить результат в виде десятичной дроби при помощи функции преобразования **evalf** (*выражение*, [*точность*]).

> **evalf(1+3/2);**

2.500000000

> **evalf(113/112);**

1.008928571

> **evalf(113/112,5);**

1.0089

> **evalf(113/112,20);**

1.0089285714285714286

По умолчанию *Maple* проводит вычисления с точностью до десятого знака после запятой, однако, задавая необязательный параметр *точность* в функции **evalf**, можно как уменьшить, так и увеличить это значение.

Если необходимо изменить точность вычислений для всех выражений, то для этого следует присвоить соответствующее значение переменной **Digits**.

> **Digits:=25:**

evalf(13/7);

1.857142857142857142857143

Кроме обычных знаков математических операций используют: ** или ^ – возведение в степень; ! – факториал; **:=** – знак присвоения; **<, >, >=, <=, =** – логические операции; % – результат предыдущей опе- рации.

Если число не помещается в одну строку, *Maple* использует символ "\" чтобы показать, что число является непрерывным.

> **100!;**

933262154439441526816992388562667004907159682643816214685929638952175999932299156089414639761565182862536979208272237582511852109168640000000000000000000000000

В *Maple* используются общепринятые названия основных математических функций.

Функция	Описание
abs	модуль
sqrt	квадратный корень
log	обыкновенный логарифм
log10	десятичный логарифм
ln	натуральный логарифм
exp	экспонента
round	округление
trunc	отсечение дробной части
Re	действительная часть
Im	комплексная часть
argument	аргумент комплексного числа

Тригонометрические функции не требуют детального описания: sin(x), cos(x), tan(x), sec(x), csc(x), cot(x), sinh(x), cosh(x), tanh(x), sech(x), csch(x), coth(x), arcsin(x), arccos(x), arctan(x), arcsec(x), arccsc(x), arccot(x), arcsinh(x), arccosh(x), arctanh(x), arcsech(x), arccsch(x), arccoth(x), arctan(y, x).

Важнейшие математические константы π и $i = \sqrt{-1}$ начинаются с больших букв. Основание натурального логарифма – ***e***, может быть получено с помощью функции **exp.**

> **Pi; evalf(%);**

$$\pi$$

$$3.14159265358979323846264338$$

> **exp(1); evalf(%);**

$$\mathbf{e}$$

$$2.71828182845904523536028747$$

> **I;**

$$I$$

> **infinity;**

$$\infty$$

Выражения можно присваивать переменным, при этом каждая переменная характеризуется типом и именем – набором символов, в которых строчные и прописные буквы различаются. Как обычно, имя не должно совпадать с существующими уже именами.

> **a:=5; A:=12; b:=15; B:=6; c:=a/b; C:=A/B;**

$$a := 5$$
$$A := 12$$
$$b := 15$$
$$B := 6$$
$$c := \frac{1}{3}$$
$$C := 2$$

2.2 ВЫЧИСЛЕНИЕ СУММЫ РЯДА, ПРОИЗВЕДЕНИЯ И ПРЕДЕЛА

Вычисление суммы членов некоторой последовательности $f(k)$ при изменении целочисленного индекса k от значения m до значения n, т.е.

$$\sum_{k=m}^{n} f(k) = f(m) + f(m+1) + \ldots + f(n-1) + f(n),$$

является достаточно распространенной операцией математического анализа. Для вычисляемой и инертной форм вычисления сумм служат функции **sum** и **Sum**.

Вычисляемая форма суммы.

> **sum(k^2,k=1..10);**

$$385$$

Инертная форма суммы.

> **Sum(k^2,k=1..10);**

$$\sum_{k=1}^{10} k^2$$

Оформление результатов расчета с использованием инертной и вычисляемой форм.

> **Sum(k^2,k=1..10)=sum(k^2,k=1..10);**

$$\sum_{k=1}^{10} k^2 = 385$$

Отметим, что если переменной-индексу (k) к моменту вычисления суммы уже присвоено какое-либо значение, то функция **sum** приведет к ошибке.

> **k:=125;**

$$k := 125$$

> **sum(k^2,k = 1 .. 10);**

Error, (in sum) summation variable previously assigned, second argument evaluates to 125 = 1 .. 10

Для того чтобы избежать ошибки следует использовать одинарные кавычки, как показано ниже.

> **sum('k^2','k'=1..10);**

$$385$$

Функция **value** служит для вычисления инертных форм.

> **S:=Sum('k^2','k'=1..10);**

$$S := \sum_{k=1}^{10} k^2$$

> **value(S);**

$$385$$

Многие бесконечные суммы сходятся к определенным значениям и *Maple* способен их вычислить.

> **Sum(1/k!, k=0..infinity)=sum(1/k!, k=0..infinity);**

$$\sum_{k=0}^{\infty} \frac{1}{k!} = \mathbf{e}$$

> **Sum(1/k^2, k=1..infinity)=sum(1/k^2,**
k=1..infinity);

$$\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Для вычисляемой и инертной форм нахождения произведений служат функции **product** и **Product.**

> **Product(k^2,k = 1 .. 5)=product(k^2,k = 1 .. 5);**

$$\prod_{k=1}^{5} k^2 = 14400$$

Для вычисления пределов служат функции **limit** и **Limit**. Вычислим предел функции $y = 12 \cdot \sin(x)$ в точке $\pi/4$.

> **Limit(12*sin(x),x=Pi/4)=limit(12*sin(x),x=Pi/4);**

$$\lim_{x \to \left(\frac{\pi}{4}\right)} 12 \sin(x) = 6\sqrt{2}$$

Предел функции $1/x$ в точке $x = 0$.

> **Limit(1/x,x=0)=limit(1/x,x=0);**

$$\lim_{x \to 0} \frac{1}{x} = \mathit{undefined}$$

Справа от нуля.

> **Limit(1/x,x=0,right)=limit(1/x,x=0,right);**

$$\lim_{x \to 0+} \frac{1}{x} = \infty$$

Слева от нуля.

> **Limit(1/x,x=0,left)=limit(1/x,x=0,left);**

$$\lim_{x \to 0-} \frac{1}{x} = -\infty$$

Первый замечательный предел.

> **Limit(sin(x)/x,x=0)=limit(sin(x)/x,x=0);**

$$\lim_{x \to 0} \frac{\sin(x)}{x} = 1$$

2.3 ОСНОВНЫЕ ТИПЫ ДАННЫХ

Рассмотрим основные типы данных, с которыми приходится встречаться при выполнении различных вычислений. Для проверки принадлежности выражения к определенному типу служат две функции:

- **whattype**(*выражение*) возвращает тип выражения;
- **type**(*выражение*, *тип*) возвращает *true* (*истина*), если *выражение* принадлежит к указанному *типу*, и *false* (*ложь*) в противном случае.

Целые

Выражение принадлежит к целому типу (тип ***integer***), если оно состоит из последовательности цифр, не разделенных между собой никакими знаками. *Maple* может работать с целыми числами практически бесконечной длины. Так, например, в *Maple 8* ограничение на длину целых чисел – 2^{28} цифр. Числа типа ***integer*** могут быть как положительными, так и отрицательными.

> **whattype(-125);**

integer

> **type(-125,integer);**

true

Дробные

Дроби (тип ***fraction***) представляются в виде: $\frac{a}{b}$, где a – целое число со знаком, b – целое число без знака. В выражении типа ***fraction*** обязательно присутствуют два поля: числитель и знаменатель, которые могут быть получены функцией **op**.

> **type(-3/7,integer);**

false

> **whattype(-3/7);**

fraction

> **op(-3/7);**

-3, 7

Числа с плавающей точкой

Числа с плавающей точкой (тип ***float***) можно определить следующим образом.

1 последовательность чисел, разделенных точкой:
   а) <*integer*>**.**<*integer*>
   б) <*integer*>**.**
   в) **.**<*integer*>
2 в виде: Float(M, E), т.е. $M*10^E$.

> **whattype(0.123);**

float

> **Float(2,3);**

2000.

Строковые типы

Выражение строкового типа (тип ***string***) – это последовательность символов, заключенных в двойные кавычки.

> **str:="Это строка !";**

str := "Это строка !"

> **whattype(str);**

string

Можно определить длину строки:

> **length(str);**

12

Из строки можно извлечь подстроку:

> **substring(str,5..10);**

"строка"

Булевы выражения

Булевы выражения (тип ***boolean***) могут принимать одно из двух значений: *true* (*истина*) или *false* (*ложь*). В булевых выражениях можно использовать следующие операторы **and**, **or**, **xor**, **implies**, **not**, а также операторы отношений **<**, **<=**, **>**, **>=**, **=**, **<>**. Функция **evalb** вычисляет сложное логическое выражение.

> **5>3;**

$3 < 5$

> **evalb(5>3);**

true

Последовательности

Последовательность (тип **exprseq**) – набор элементов, разделенных запятыми, без скобок.

> **S:=1,2,3,4,5,6,7,8,9,10;**

$S := 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

> **whattype(S);**

exprseq

Для генерации последовательностей служит функция **seq:**

> **S:=seq(i,i=1..10);**

$S := 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

> **Q:=seq(i^2,i=1..10);**

$Q := 1, 4, 9, 16, 25, 36, 49, 64, 81, 100$

Последовательность можно также получить при помощи оператора формирования последовательности – **$**.

> **$ 2..5;**

2, 3, 4, 5

> **i^2 $ i = 2/3 .. 8/3;**

$\frac{4}{9}, \frac{25}{9}, \frac{64}{9}$

> **a[i] $ i = 1..3;**

a_1, a_2, a_3

> **x$4;**

x, x, x, x

Пустая последовательность обозначается **NULL**.

Множества

Множество (тип **set**) – набор элементов, разделенных запятыми и заключенный в фигурные скобки.

Для множеств действительны все правила преобразования, принятые в классической математике.

Задание множества и определение числа элементов.

> **{1, 2, 3, 4, 5, 1};**

$$\{1, 2, 3, 4, 5\}$$

> **whattype(%);**

set

Количество элементов во множестве.

> **nops(%%);**

5

Как видно из предыдущего примера, множество не может содержать два одинаковых элемента.
Объединение множеств

> **{1, 2, 3, 4} union {3, 4, 5, 6};**

$$\{1, 2, 3, 4, 5, 6\}$$

Пересечение множеств

> **{1, 2, 3, 4} intersect {3, 4, 5, 6};**

$$\{3, 4\}$$

Вычитание множеств

> **{1, 2, 3, 4} minus {3, 4, 5, 6};**

$$\{1, 2\}$$

Множества могут состоять не только из чисел

> **U:={a, b, c, d, e, f};**

$$U := \{a, b, c, d, e, f\}$$

Извлечение с 2 по 4 элемент из множества *U*.

> **op(2..4,U);**

b, *c*, *d*

Проверка принадлежности элемента множеству *U*.

> **member(a,U);**

true

> **member(g,U);**

false

Задание множества в виде последовательности

> **P:={seq(a[i],i=1..5)};**

$$P := \{a_1, a_2, a_3, a_4, a_5\}$$

Добавление элемента к множеству *P*

> **P:={op(P),a[6]};**

$$P := \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

Удаление третьего элемента из множества *P*

> **P:=subsop(3=NULL,P);**

$$P := \{a_1, a_2, a_4, a_5, a_6\}$$

Списки

Список (тип **list**) – набор элементов, разделенных запятыми и заключенный в квадратные скобки.

> **[a, b, c, d];**

$$[a, b, c, d]$$

> **whattype(%);**

list

Со списками можно проводить математические операции, например, дифференцирование:

> **L:=[sin, cos, tan];**

$$L := [\sin, \cos, \tan]$$

> **D(L);**

$$[\cos, -\sin, 1 + \tan^2]$$

Задание списка в виде последовательности.

> **[seq(x[i],i=1..5)];**

$$[x_1, x_2, x_3, x_4, x_5]$$

Список, в отличие от множества, может содержать одинаковые элементы.

> **[1, 2, 3, 4, 5, 2];**

$$[1, 2, 3, 4, 5, 2]$$

Во множестве порядок следования элементов не имеет значения, а в списке он существенен:

> **evalb({a,b,c}={c,b,a});**

true

> **evalb([a,b,c]=[c,b,a]);**

false

Массивы

Массив (тип **array**) – конечный список с целочисленными индексами. Для создания массива служит функция **array**.

Создаем пустой массив из пяти элементов, заполняем его в цикле **for** квадратами индексов и выводим на печать функцией **print**:

> **A:=array(1..5);**

$$A := \text{array}(1 .. 5, [\ ])$$

> **whattype(%);**

array

> **for i from 1 to 5 do A[i]:=i^2 end do;**

$$A_1 := 1$$
$$A_2 := 4$$
$$A_3 := 9$$
$$A_4 := 16$$
$$A_5 := 25$$

> **print(A);**

$$[1, 4, 9, 16, 25]$$

Создаем двухмерный массив 2 × 2 и сразу присваиваем значения

> **B:=array(1..2, 1..2, [[1, 3], [1/2, 5]]);**

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

Массив в *Maple* может содержать элементы разных типов.

> **C:=array(1..2, 1..2, [[x^3, 3], [sin(x), 2.33]]);**

$$C := \begin{bmatrix} x^3 & 3 \\ \sin(x) & 2.33 \end{bmatrix}$$

Функция **map** позволяет выполнять какую-либо операцию над всеми элементами массива.
Дифференцирование по *x* всех элементов массива *C*

> **map(diff,C,x);**

$$\begin{bmatrix} 3x^2 & 0 \\ \cos(x) & 0 \end{bmatrix}$$

Извлечение квадратного корня из всех элементов массива *C*

> **map(sqrt,C);**

$$\begin{bmatrix} \sqrt{x^3} & \sqrt{3} \\ \sqrt{\sin(x)} & 1.526433752 \end{bmatrix}$$

Таблицы

В отличие от массива, где индексы – целочисленные значения, расположенные по порядку номеров, индексы у таблицы (тип **table**) – любые значения.

Если индексы не определены, то *Maple* присваивает по порядку целочисленные индексы

> **A:=table([Иванов,Петров,Сидоров]);**

$$A := \text{table}([1 = Иванов, 2 = Петров, 3 = Сидоров])$$

> **whattype(%);**

table

> **A[2];**

Петров

Индексы таблицы можно присваивать произвольно.

> **B:=table([(первый)=Иванов, (второй)=Петров, (третий)=Сидоров]);**

$$B := \text{table}([первый = Иванов, второй = Петров, третий = Сидоров])$$

> **B[второй];**

Петров

2.4 ОПЕРАЦИИ С ФОРМУЛАМИ

При работе математическими выражениями приходится выполнять такие операции, как приведение подобных членов, раскрытие скобок, разложение на множители. В пакете *Maple* это можно сделать при помощи специальных функций.

Функция	Описание
simplify	упростить выражение
factor	факторизовать
expand	разложить (раскрыть все скобки)
normal	привести выражение к "нормальному" виду
convert	переписать в заданном виде
coeff	выделить коэффициенты полинома

collect	собрать вместе части выражения

Упрощение выражений.

> **sin(x)^2+cos(x)^2;**

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

> **simplify(%);**

$$1$$

> **cos(x)^5+sin(x)^4+2*cos(x)^2-2*sin(x)^2-cos(2*x);**

$$\cos(x)^5 + \sin(x)^4 + 2\cos(x)^2 - 2\sin(x)^2 - \cos(2x)$$

> **simplify(%);**

$$\cos(x)^4 (\cos(x) + 1)$$

> **sqrt(x^2);**

$$\sqrt{x^2}$$

> **simplify(%);**

$$\mathrm{csgn}(x)\, x$$

В последнем примере функция **csgn** возвращает знак действительного или комплексного числа

$$c\,\mathrm{sgn}(x) = \begin{cases} 1 & \text{если } \mathrm{Re}(x) > 0 \text{ или } \mathrm{Re}(x) = 0 \text{ и } \mathrm{Im}(x) > 0 \\ -1 & \text{если } \mathrm{Re}(x) < 0 \text{ или } \mathrm{Re}(x) = 0 \text{ и } \mathrm{Im}(x) < 0 \end{cases}.$$

Факторизировать – значит разложить выражение на множители.

> **6*x^2+18*x-24;**

$$6x^2 + 18x - 24$$

> **factor(%);**

$$6 (x + 4) (x - 1)$$

Для разложения на множители целых чисел служит функция **ifactor**.

> **ifactor(132);**

$$(2)^2 (3) (11)$$

Раскрытие скобок.

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

$$(x + 1) (x + 2)$$

> **expand(%);**

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

> **(x+1)/(x+2);**

$$\frac{x+1}{x+2}$$

> **expand(%);**

$$\frac{x}{x+2} + \frac{1}{x+2}$$

Функция **normal** обычно используется для полиномов и рациональных функций, хотя иногда применима и для более общих выра- жений.

> **1/x+x/(x+1);**

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

> **normal(%);**

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

> **normal(%%, expanded);**

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

В последнем примере дополнительно происходит раскрытие скобок.

> **sin(x)+1/sin(x)^2;**

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

> **normal(%);**

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

Функция **convert** преобразует выражение в различные формы.

Преобразование чисел в различные системы счисления:

> **convert(1019, binary);**

$$1111111011$$

> **convert(1019, hex);**

$$3FB$$

Преобразование десятичной дроби в натуральную.

> **convert(1.23456, fraction);**

$$\frac{3858}{3125}$$

Преобразование выражения в элементарные дроби.

> **(x^3+x)/(x^2-1);**

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

> **convert(%, parfrac, x);**

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

Преобразование экспоненциального выражения к тригонометрическому виду.

> **1/4*exp(x)^2-1/4/exp(x)^2;**

$$\frac{1}{4}(\mathbf{e}^x)^2 - \frac{1}{4}\frac{1}{(\mathbf{e}^x)^2}$$

> **convert(%,trig);**

$$\frac{1}{4}(\cosh(x) + \sinh(x))^2 - \frac{1}{4}\frac{1}{(\cosh(x) + \sinh(x))^2}$$

И обратно в экспоненциальный.

> **convert(%,exp);**

$$\frac{1}{4}(\mathbf{e}^x)^2 - \frac{1}{4}\frac{1}{(\mathbf{e}^x)^2}$$

Выделение коэффициентов полинома осуществляется функцией **coeff** (*выражение, переменная, степень*).

> **p := 2*x^2 + 3*y^3 - 5;**

$$p := 2x^2 + 3y^3 - 5$$

> **coeff(p,x,0);**

$$3y^3 - 5$$

> **coeff(p,x,1);**

$$0$$

> **coeff(p,x,2);**

$$2$$

Функция **collect** позволяет собирать вместе коэффициенты при одинаковых степенях. В следующих примерах собираются коэффициенты при $\ln(x)$ и x.

> **a*ln(x)-ln(x)*x-x;**

$$a\ln(x) - \ln(x)\,x - x$$

> **collect(%,ln(x));**

$$(a - x)\ln(x) - x$$

> **y/x+2*z/x+x^(1/3)-y*x^(1/3);**

$$\frac{y}{x} + \frac{2z}{x} + x^{(1/3)} - y\,x^{(1/3)}$$

> **collect(%,x);**

$$(1 - y)\,x^{(1/3)} + \frac{y + 2z}{x}$$

Функция **trigsubs** выдает все тригонометрические эквиваленты выражения в виде списка.

> **trigsubs(sin(alpha+beta));**

$$\left[\sin(\alpha+\beta),\ -\sin(-\alpha-\beta),\ 2\sin\left(\frac{\alpha}{2}+\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}+\frac{\beta}{2}\right),\ \frac{1}{\csc(\alpha+\beta)},\right.$$

$$\left.-\frac{1}{\csc(-\alpha-\beta)},\ \frac{2\tan\left(\frac{\alpha}{2}+\frac{\beta}{2}\right)}{1+\tan\left(\frac{\alpha}{2}+\frac{\beta}{2}\right)^2},\qquad \frac{-1}{2}I\left(\mathbf{e}^{((\alpha+\beta)I)} - \mathbf{e}^{(-I(\alpha+\beta))}\right)\right]$$

Напомним, что извлекать элементы из списка можно при помощи функции **op**, указав первым параметром номер элемента. Извлечем третий элемент из последнего списка.

> **op(3,%);**

$$2\sin\left(\frac{\alpha}{2}+\frac{\beta}{2}\right)\cos\left(\frac{\alpha}{2}+\frac{\beta}{2}\right)$$

2.5 ПРОИЗВОДНЫЕ И ИНТЕГРАЛЫ

Вычисление производных осуществляется с помощью функций **diff** и **Diff**. Первым параметром этих функций является дифференцируемое выражение, далее – имя переменной или последовательность имен переменных.

Дифференцируем функцию sin(x) по x.

> **diff(sin(x),x);**

$$\cos(x)$$

И второй раз.

> **diff(%,x);**

$$-\sin(x)$$

Но двойное дифференцирование можно выполнить так

> **diff(sin(x),x,x);**

$$-\sin(x)\ .$$

Или так

> **diff(sin(x),x$2);**

$$-\sin(x)\ .$$

В последнем примере для задания последовательности из двух переменных x используется оператор генерации последовательности – **$**. Одна функция **diff** может выполнить дифференцирование по нескольким переменным.

> **Diff(y*sin(x)/cos(y),x,y) = diff(y*sin(x)/cos(y),x,y);**

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

Для вычисления интегралов используются функции **int** и **Int**. При вычислении неопределенных интегралов первый параметр – интегрируемое выражение, второй – имя переменной.

> **Int(sin(x),x)=int(sin(x),x);**

$$\int \sin(x)\,dx=-\cos(x)$$

> **Int(x/(x^3-1),x)=int(x/(x^3-1),x);**

$$\int \frac{x}{x^3-1}\,dx=-\frac{1}{6}\ln(x^2+x+1)+\frac{1}{3}\sqrt{3}\,\arctan\left(\frac{(2x+1)\sqrt{3}}{3}\right)+\frac{1}{3}\ln(x-1)$$

> **eq:=exp(-x^2)*ln(x);**

$$eq := \mathbf{e}^{(-x^2)}\ln(x)$$

> **int(eq,x);**

$$\int \mathbf{e}^{(-x^2)}\ln(x)\,dx$$

Если интеграл не берется, как в последнем примере, то подынтегральное выражение может быть разложено в степенной ряд функцией **series**. Разложим подынтегральное выражение в ряд до 8-го порядка.

> **series(eq,x,8);**

$$\ln(x)-\ln(x)\,x^2+\frac{1}{2}\ln(x)\,x^4-\frac{1}{6}\ln(x)\,x^6+\mathrm{O}(x^8)$$

Теперь можно интегрировать ряд.

> **int(%,x);**

$$x\ln(x)-x-\frac{1}{3}\ln(x)\,x^3+\frac{x^3}{9}+\frac{1}{10}\ln(x)\,x^5-\frac{x^5}{50}-\frac{1}{42}\ln(x)\,x^7+\frac{x^7}{294}+\mathrm{O}(x^9)$$

При вычислении определенных интегралов необходимо задать пределы интегрирования.

> **Int(sin(x),x=0..Pi/4)=int(sin(x),x=0..Pi/4);**

$$\int_0^{\frac{\pi}{4}} \sin(x)\,dx=-\frac{\sqrt{2}}{2}+1$$

Численное вычисление определенного интеграла.

> **evalf(Int(sin(x),x=0..Pi/4));**

0.2928932188

Вычисление интеграла с бесконечным верхним пределом.

> **Int(exp(-x),x=0..infinity)=int(exp(-x), x=0..infinity);**

$$\int_0^{\infty} \mathbf{e}^{(-x)}\, dx = 1$$

2.6 ПАКЕТЫ РАСШИРЕНИЙ И РАБОТА С НИМИ

Некоторые функции *Maple* помимо ядра могут находиться в пакетах расширений, входящих в базовую поставку системы. Перед использованием таких функций их надо загрузить. Для загрузки всех функций какого-либо пакета используется функция **with** (*имя_пакета*). Для загрузки избранных функций пакета – **with** (*имя_пакета, функция_1, функция_2, …*). Отметим, что некоторые функции пакетов расширений могут переопределять одноименные функции ядра.

Пакет **linalg** содержит более ста функций для решения задач линейной алгебры. Рассмотрим некоторые из них на примере двух матриц 3 × 3 созданных при помощи функции **matrix**, аналогичной функции **array**.

> **with(linalg):**
> **A := matrix(3,3,[[1,2,3],[4,5,6],[7,8,9]]) ;**

$$A := \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

> **B := matrix(3,3,[[7,4,3],[1,2,5],[8,9,6]]);**

$$B := \begin{bmatrix} 7 & 4 & 3 \\ 1 & 2 & 5 \\ 8 & 9 & 6 \end{bmatrix}$$

Вычислим детерминант (определитель) матриц ***A*** и ***B***.

> **det(A);**

0

> **det(B);**

-116

Сумма матриц

> **matadd(A,B);**

$$\begin{bmatrix} 8 & 6 & 6 \\ 5 & 7 & 11 \\ 15 & 17 & 15 \end{bmatrix}$$

Произведение матриц

> **multiply(A, B);**

$$\begin{bmatrix} 33 & 35 & 31 \\ 81 & 80 & 73 \\ 129 & 125 & 115 \end{bmatrix}$$

Транспонирование матрицы ***A***

> **transpose(A);**

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$$

Функция **multiply** умножает матрицы, а **inverse** находит обратную матрицу.

> **multiply(B, inverse(B));**

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Рассмотрим некоторые функции пакета **student**, который позволяет проводить вычисления поэтапно, что может быть особенно полезно студентам.

Функция **intparts** – интегрирование по частям.

> **with(student):**
> **Int(x*cos(x),x);**

$$\int x \cos(x)\, dx$$

> **intparts(Int(x*cos(x),x),x);**

$$x \sin(x) - \int \sin(x)\, dx$$

> **value(%);**

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

Интегрирование подстановкой – функция **changevar**.

> **Int((cos(x)+1)^3*sin(x), x);**

$$\int (\cos(x) + 1)^3 \sin(x)\, dx$$

> **changevar(cos(x)+1=u, Int((cos(x)+1)^3*sin(x), x), u);**

$$\int -u^3\, du$$

> **Int(sqrt(1-x^2), x=a...b);**

$$\int_a^b \sqrt{1 - x^2}\, dx$$

> **changevar(x=sin(u), Int(sqrt(1-x^2), x=a...b), u);**

$$\int_{\arcsin(a)}^{\arcsin(b)} \sqrt{1 - \sin(u)^2}\, \cos(u)\, du$$

Заметим, что в последнем примере пределы интегрирования изменились автоматически.

Для вычисления двойных и тройных интегралов служат функции **Doubleint** и **Tripleint**.

> **Tripleint((r^2*sin(f),r=0..4*cos(f),f=0..Pi/4, t=0..Pi/2));**

$$\int_0^{\frac{\pi}{2}} \int_0^{\frac{\pi}{4}} \int_0^{4\cos(f)} r^2 \sin(f)\, dr\, df\, dt$$

> **value(%);**

$$2\,\pi$$

Функции **simpson** и **trapezoid** реализуют числовое приближение к интегралу методами Симпсона и трапеций.

> **simpson(x^k*ln(x), x=1..3);**

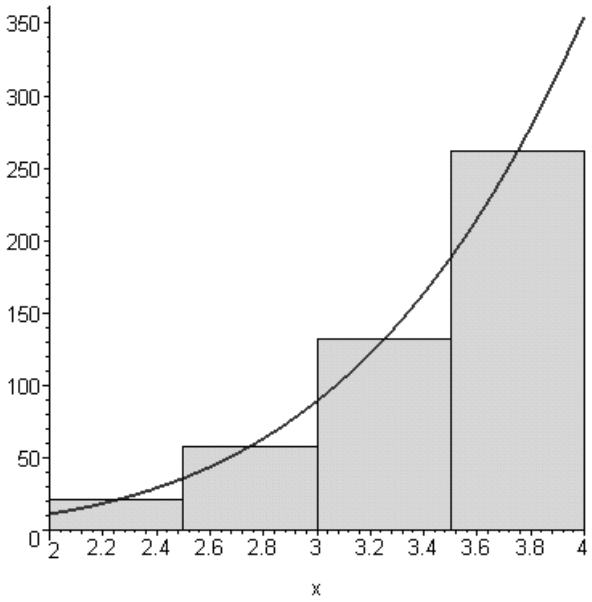
$$\frac{1}{6}3^k \ln(3) + \frac{2}{3}\left(\sum_{i=1}^{2}\left(\frac{1}{2}+i\right)^k \ln\left(\frac{1}{2}+i\right)\right) + \frac{1}{3}\left(\sum_{i=1}^{1}(1+i)^k \ln(1+i)\right)$$

> **trapezoid(x^k*ln(x), x=1..3);**

$$\frac{1}{2}\left(\sum_{i=1}^{3}\left(1+\frac{i}{2}\right)^k \ln\left(1+\frac{i}{2}\right)\right) + \frac{1}{4}3^k \ln(3)$$

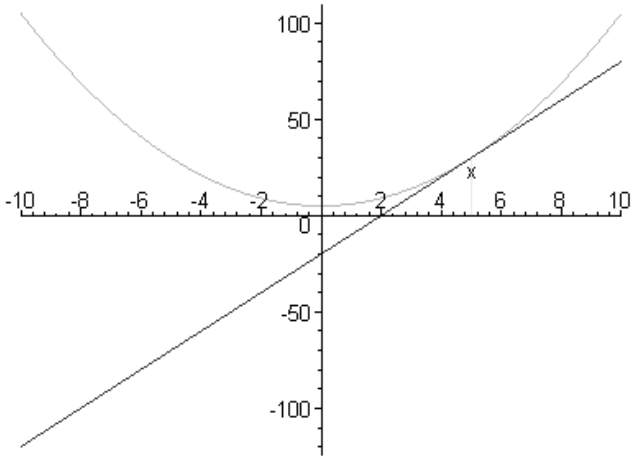
Функции **leftbox**, **middlebox** и **rightbox** иллюстрируют интегрирование методами левых, центральных и правых прямоугольников.

> **middlebox(x^4*ln(x), x=2..4);**



Функция **showtangent** строит график функции и касательную в указанной точке.

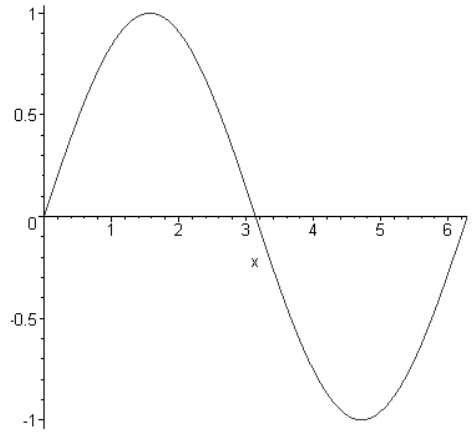
> **showtangent(x^2+5, x = 5);**



3 ДВУХМЕРНАЯ ГРАФИКА

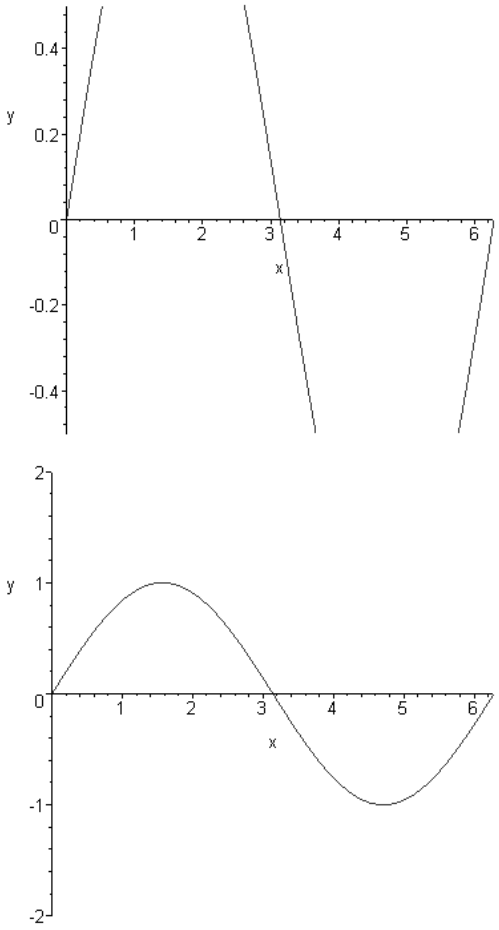
Основной функцией построения графиков является **plot**.

> **plot(sin(x),x=0..2*Pi);**



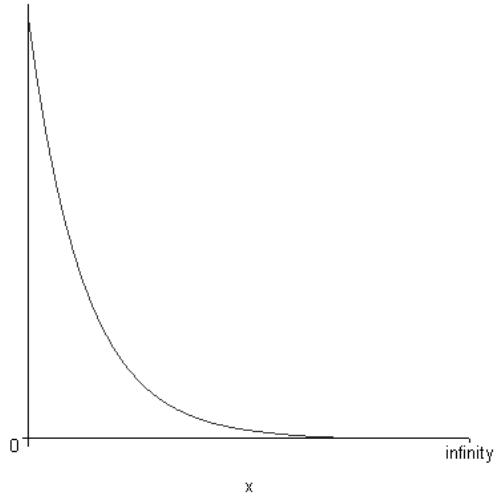
Кроме самой функции, график которой нужно построить, обязательным параметром является область. *Область* – это окно декартовой системы координат, в котором строится график. Если в области задан только диапазон по *x* (как в предыдущем примере), то диапазон по *y* рассчитывается автоматически.

> **plot(sin(x),x=0..2*Pi,y=-0.5..0.5);**
plot(sin(x),x=0..2*Pi,y=-2..2);



Области можно задавать с использованием констант, в том числе *infinity*.

> **plot(exp(-x),x=0..infinity);**

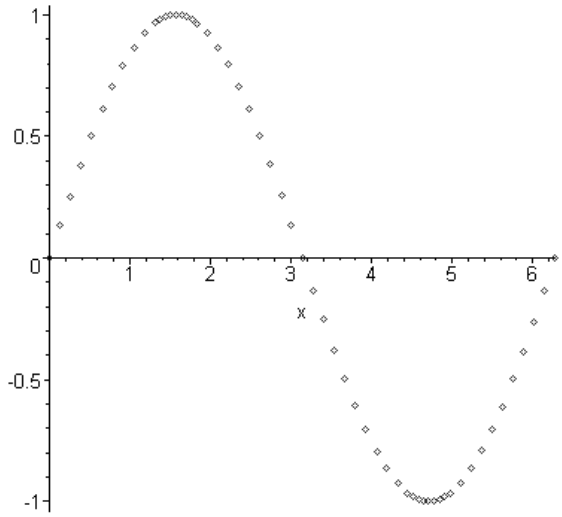


Функция **plot** может иметь 27 дополнительных параметров; некоторые из них описаны ниже.

При построении графиков можно выбирать *стиль* интерполирования. Стиль задается с помощью ключевого слова *style*. Существуют три стиля:

- POINT – график строится по точкам;
- LINE – точки соединяются прямыми. Используется по умолчанию;
- PATCH – применяется для построения раскрашенных многоугольников.

> **plot(sin(x),x=0..2*Pi,style=POINT);**



Тип линии может быть задан параметром *linestyle*. Доступны следующие стили:

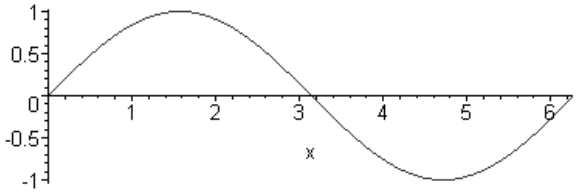
- SOLID – сплошная линия;
- DOT – линия из точек;
- DASH – штриховая линия;
- DASHDOT – штрихпунктирная линия.

Цвет линии задается параметром *color*, толщина линии параметром *thickness*.

> **plot(sin(x),x=0..2*Pi,linestyle=DASH,**
color=blue,thickness=3);

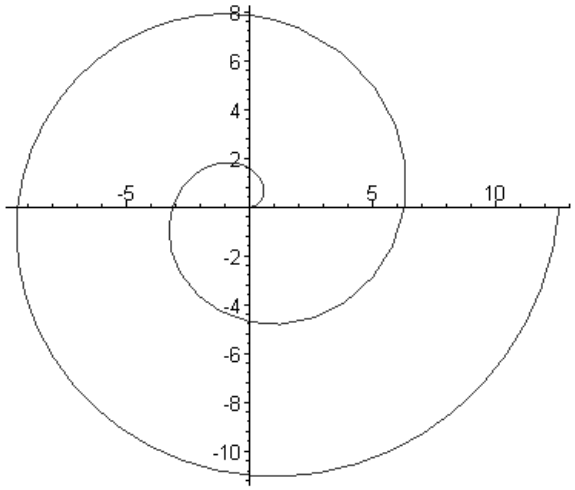
При выводе графиков *Maple* выбирает масштабы по осям автоматически так, чтобы график был наиболее информативен, но, используя параметр *scaling* (масштабирование), можно запретить использование различных масштабов по осям, как это сделано в следующем примере.

> **plot(sin(x),x=0..2*Pi,scaling=CONSTRAINED);**



Параметр *coords* позволяет выбрать систему координат. По умолчанию *Maple* строит графики в декартовой системе координат. В следующем примере график функции $y = x$ построен в полярной системе координат.

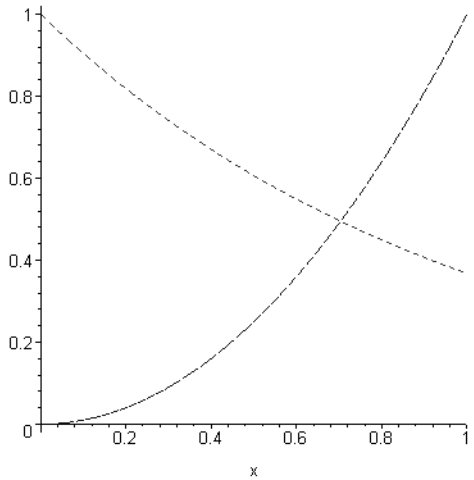
> **plot(x,x=0..4*Pi,coords=polar, scaling=CONSTRAINED);**



3.1 СОВМЕЩЕНИЕ ГРАФИКОВ

Maple может построить несколько графиков на одной координатной плоскости. Для этого достаточно указать в функции **plot** множество или список функций, при этом для разных графиков автоматически выбираются различные цвета. При необходимости для каждой функции можно указать желаемый цвет и стиль построения.

> **plot([x^2,exp(-x)],x=0..1,color=[blue,violet],**
linestyle=[DASH,DASHDOT]);



3.2 АНИМАЦИЯ ГРАФИКОВ

Функция **animate** позволяет создавать анимированные изображения. Эта функция находится в пакете *plots*, который предварительно должен быть подключен. Суть анимации заключается в построении серии изображений, причем каждое изображение (фрейм) связано с изменяемой во времени переменной *t*.

> **with(plots):**
animate(sin(x*t),x=-10..10,t=1..2);

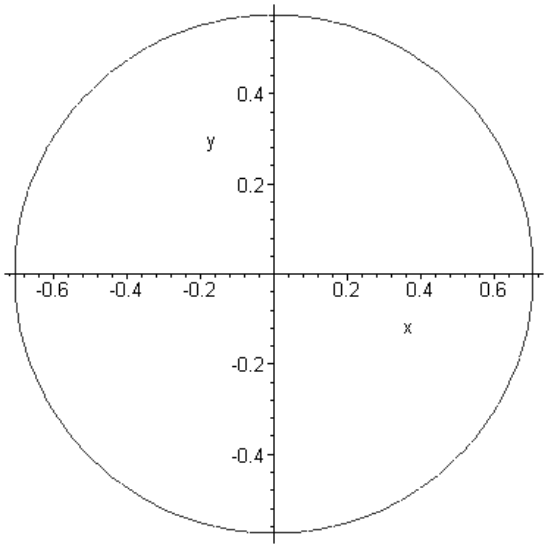
При выделении полученного изображения появляется панель проигрывания анимационных клипов. Она имеет кнопки управления с обозначениями, принятыми у магнитофонов.

Нажав кнопку  , можно наблюдать анимированное изображение.

3.3 ПОСТРОЕНИЕ ГРАФИКА НЕЯВНОЙ ФУНКЦИИ

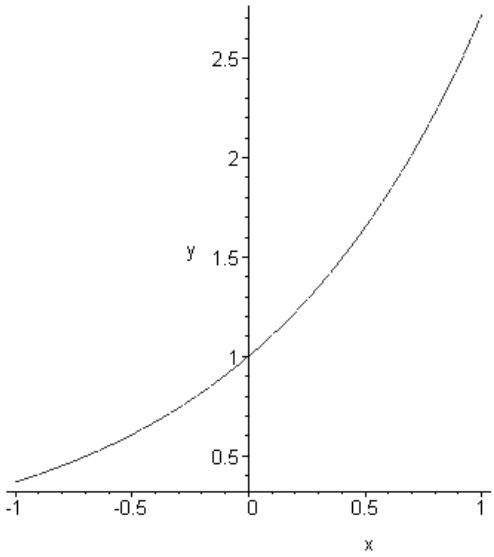
Для построения графика функции заданной неявно служит функция **implicitplot** из пакета *plots*.

> **with(plots):**
implicitplot(2*x^2 + 3*y^2 = 1,x=-1..1,y=-1..1);



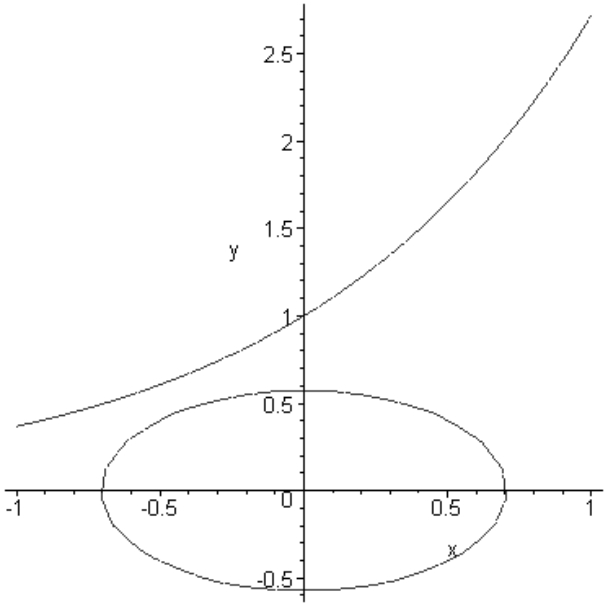
С помощью **implicitplot** можно также строить графики явно заданных функций.

> **implicitplot(y = exp(x),x=-1..1,y=-1..3);**



Указав в качестве первого параметра функции **implicitplot** множество функций, можно совместить графики в одной координатной плоскости.

> **implicitplot({2*x^2 + 3*y^2 = 1,y = exp(x)}, x=-1..1,y=-1..3);**

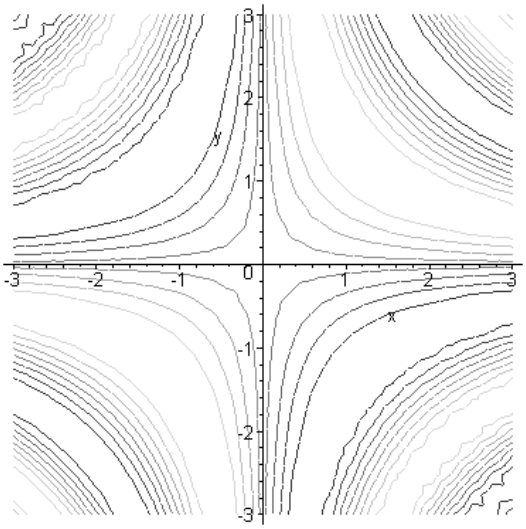


3.4 ПОСТРОЕНИЕ ГРАФИКОВ ЛИНИЯМИ РАВНОГО УРОВНЯ

Линии равного уровня образуются, если мысленно рассечь трехмерную поверхность плоскостями, параллельными плоскости *XY*. Линии пересечение этих плоскостей с трехмерной поверхностью и являются линиями равного уровня.

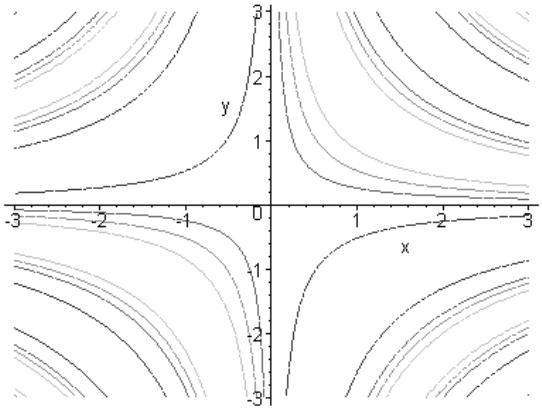
Для построения таких графиков используется функция **contourplot** из пакета *plots*.

> **with(plots):**
contourplot(sin(x*y),x=-3..3,y=-3..3);



В качестве дополнительных параметров может быть задан размер сетки (*grid*), которая будет использована для построения графиков и значения функции, которые образуют линии равного уровня (*contours*).

> **contourplot(sin(x*y),x=-3..3,y=-3..3, grid=[50,50], contours=[-1/2,1/4,1/2,3/4]);**

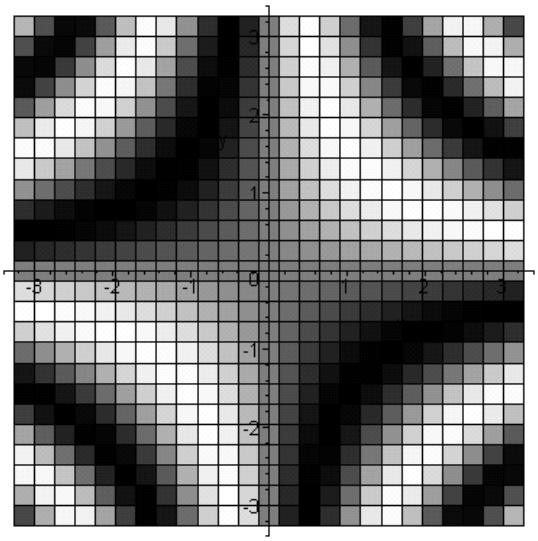


Опция *filled = true* обеспечивает окрашивание замкнутых областей, образованных линиями равного уровня, что придает графику большую выразительность.

3.5 ГРАФИК ПЛОТНОСТИ

Иногда трехмерные поверхности необходимо отобразить на плоскости как графики плотности окраски – чем выше высота поверхности, тем плотнее окраска. Такой вид графиков создается функцией **densityplot**, которая также входит в пакет *plots*.

> **with(plots):**
densityplot(sin(x*y),x=-Pi..Pi,y=-Pi..Pi, grid=[25,25]);

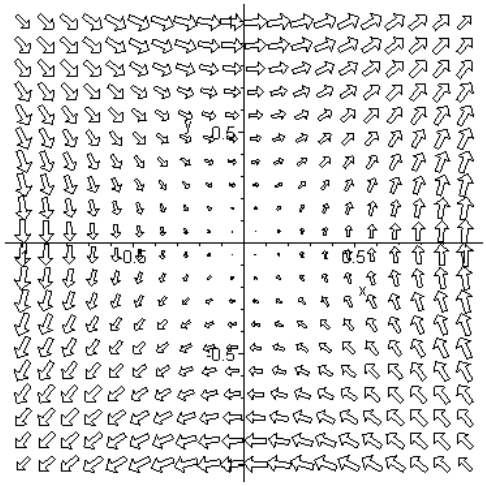


Создаваемое изображение разбито на прямоугольники, количество которых определяет параметр *grid*.

3.6 ГРАФИК ВЕКТОРНОГО ПОЛЯ ГРАДИЕНТА

Функция пакета *plots* – **gradplot**, служит для отображения векторного поля градиента. Параметр *arrow* позволяет изменять форму стрелок.

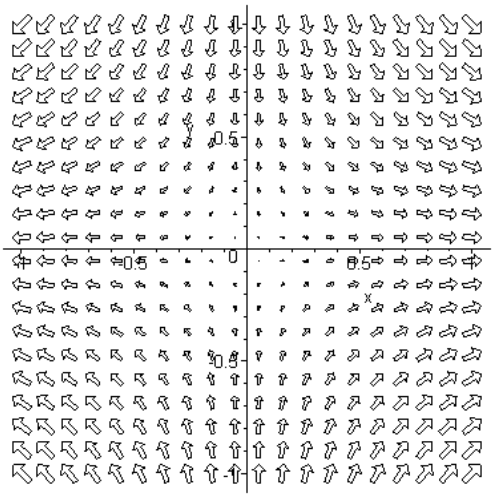
> **with(plots):**
gradplot(sin(x*y),x=-1..1,y=-1..1,arrows=THICK);



3.7 ГРАФИК ВЕКТОРНОГО ПОЛЯ

Функция пакета *plots* – **fieldplot**, применяется для отображения полей. Особенность таких графиков в том, что для их построения используются стрелки, направление которых соответствует направлению изменения градиента поля, а длина – значению градиента.

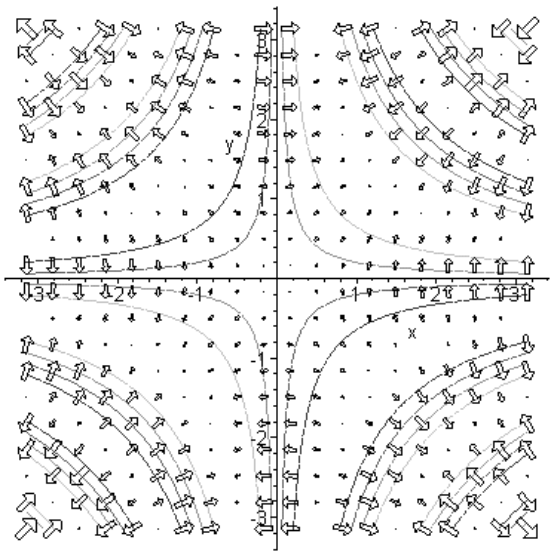
> **with(plots):**
fieldplot([x/(x^2+y^2+4)^(1/2),
-y/(x^2+y^2+4)^(1/2)], x=-1..1,y=-1..1, arrows=THICK);



3.8 СОВМЕЩЕНИЕ ГРАФИКОВ ПОСТРОЕННЫХ РАЗЛИЧНЫМИ ФУНКЦИЯМИ

Часто бывает необходимо совместить на одной координатной плоскости графики, построенные разными функциями. Это может быть сделано с помощью функции **display**.

```
> with(plots):
Ris_1:=gradplot(sin(x*y),x=-Pi..Pi,y=-Pi..Pi, arrows=THICK):
Ris_2:=contourplot (sin(x*y),x=-Pi..Pi,y=-Pi..Pi, grid=[50,50],contours=4):
display(Ris_1,Ris_2);
```



Функция **display** с параметром *insequence* = *true* (по умолчанию *insequence* = *false*) отображает последовательности, множества или списки с графическими данными, создавая структуру, аналогичную **animate**. Фреймы показываются друг за другом, что во многих случаях позволяет получить эффект анимации.

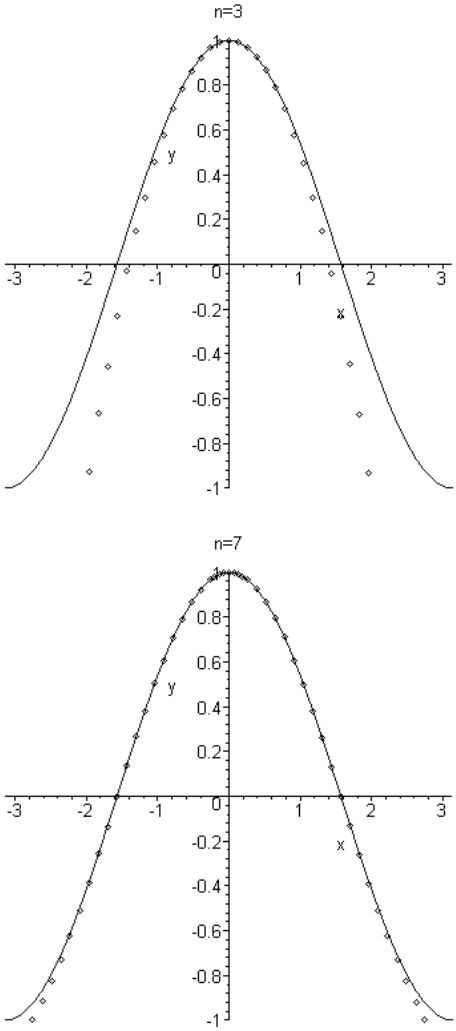
В следующем примере создается последовательность, содержащая графические данные, и осуществляется их показ посредством функции **display.** Каждый фрейм содержит график функции cos(*x*) и разложение этой функции в степенной ряд *n*-ой степени в точке *x* = 0. Графики строятся на интервале от –π до π.

```
> f:=cos(x);
L:=seq(plot([f,convert(series(f,x=0,n),polynom)], x=-Pi..Pi,   y=-1..1, style=[line, point], color=[blue,
black],title=cat("n=",n)
),                  # end plot
```

n=1..10): # end seq

plots[display](L, insequence=true);

$$f := \cos(x)$$



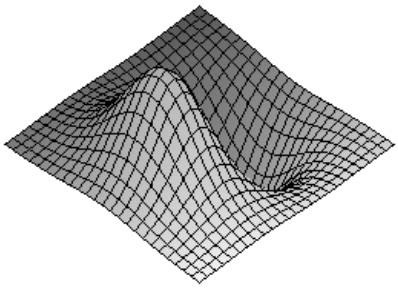
Из-за невозможности передать эффект анимации на бумаге, приведем только два фрейма, соответствующие степени ряда 3 и 7.

Отметим, что в последней строке примера использован вызов функции **display** из пакета *plots* без загрузки всех функций пакета.

4 ТРЕХМЕРНАЯ ГРАФИКА

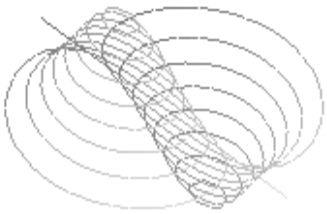
Maple имеет большое количество функций трехмерной графики. Многие функции трехмерной графики аналогичны ранее рассмотренным функция двухмерной графики. Основная функция трехмерной графики – **plot3d**.

> **plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2, grid=[25,25]);**



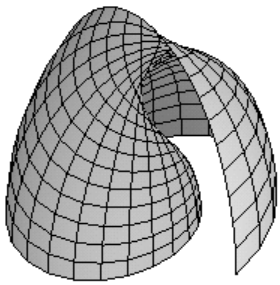
Использование параметра *style = CONTOUR* в функции **plot3d** позволяет построить линии равного уровня.

> **plot3d(x*exp(-x^2-y^2),x=-2..2,y=-2..2, grid=[25,25], style=CONTOUR);**



Функция **plot3d** может построить график функции заданной параметрически.

> **plot3d([x*sin(x)*cos(y),x*cos(x)*cos(y),x*sin(y)], x=0..2*Pi,y=0..Pi);**



Основные функции трехмерной графики пакета *plots*

ункция	**Назначение**
contour-plot3d	Строит линии равного уровня. Идентична функции **plot3d** с параметром *style = CONTOUR*
gradplot3d	Строит трехмерное поле градиента
fieldplot3d	Строит трехмерное векторное поле
implicit-	Строит неявно заданную функцию трех пе-

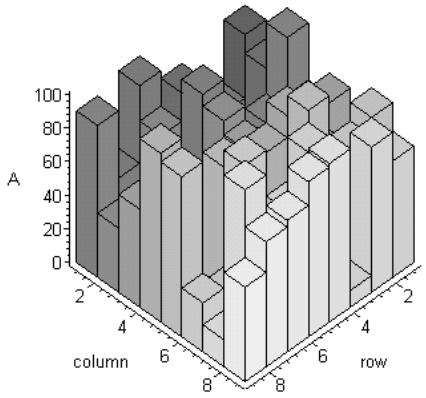
plot3d	ременных
matrixplot	Строит поверхность, заданную таблицей
cylinder-plot	Строит поверхность, заданную в цилиндрических координатах
sphereplot	Строит поверхность, заданную в сферических координатах
spacecurve	Строит кривую в трехмерном пространстве. Кривая должна быть задана параметрически
surfdata	Строит поверхность, проходящую через заданные точки
tuberplot	Строит поверхность, определяемую параметрически заданной пространственной кривой и радиусом
display3d	Выводит трехмерные графические структуры в общих осях координат

Примеры использования функций трехмерной графики

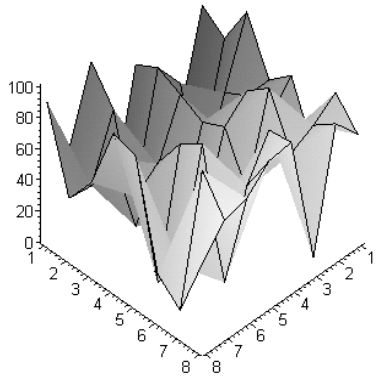
Заполним массив *A* размером 8 × 8 случайными числами. Выведем его на печать, построим трехмерную гистограмму и поверхность.

```
> with(plots):
rnd:=rand(1..100):
A:=array(1..8,1..8,[]):
for i from 1 to 8 do
   for j from 1 to 8 do
        A[i,j]:= rnd():
   end do:
end do:
print(A);
matrixplot(A,heights=histogram,axes=frame);
```

$$\begin{bmatrix} 82 & 71 & 98 & 64 & 77 & 39 & 86 & 69 \\ 22 & 10 & 56 & 64 & 58 & 61 & 75 & 86 \\ 17 & 62 & 8 & 50 & 87 & 99 & 67 & 10 \\ 74 & 82 & 75 & 67 & 74 & 43 & 92 & 94 \\ 1 & 12 & 39 & 14 & 21 & 45 & 66 & 92 \\ 96 & 75 & 10 & 61 & 83 & 93 & 14 & 78 \\ 50 & 36 & 62 & 49 & 4 & 24 & 96 & 74 \\ 90 & 38 & 58 & 100 & 95 & 29 & 16 & 56 \end{bmatrix}$$



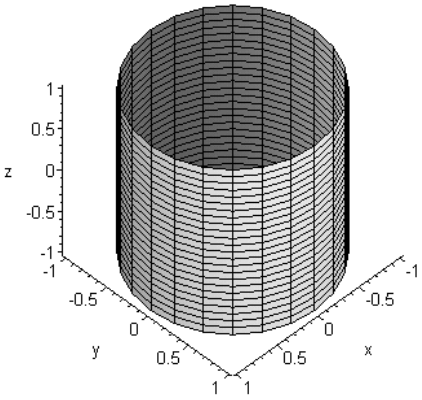
> **S:=[seq([seq([i,j,A[i,j]],j=1..8)],i=1..8)]:**
surfdata(S,axes=frame);



В этом примере двухмерный массив *A* заполняется посредством двух вложенных циклов, а функция *Maple* **rand (1 … 100)** создает процедуру *rnd* ( ), генерирующие случайные числа в диапазоне от 1 до 100. Функция **matrixplot** с параметром *heights = histogram* строит трехмерную гистограмму, а **surfdata** – поверхность по точкам, находящимся в списке *S*.

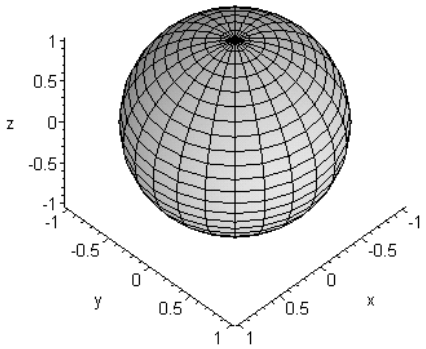
Использование функции **cylinderplot**.

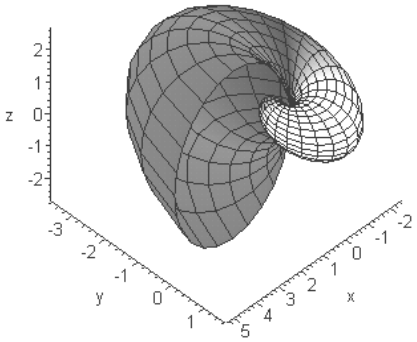
> **with(plots):**
cylinderplot(1,theta=0..2*Pi,z=-1..1, axes=frame);



Использование функции **sphereplot**.

> **with(plots):**
sphereplot(1,theta=0..2*Pi,phi=0..Pi,axes=frame);
sphereplot((1.3)^z*sin(theta),z=-1..2*Pi, theta=0..Pi, style=patch,color=z,axes=frame);





5 РЕШЕНИЕ УРАВНЕНИЙ, СИСТЕМ УРАВНЕНИЙ И НЕРАВЕНСТВ

Для аналитического решения линейных и нелинейных уравнений и систем служит функция **solve**, например:

> **solve(a*x^2+b*x+c=0,x);**

$$\frac{-b+\sqrt{b^2-4\,a\,c}}{2\,a}, \frac{-b-\sqrt{b^2-4\,a\,c}}{2\,a}$$

В качестве первого параметра записано уравнение, а второго – переменная, относительно которой уравнение следует решать. Если правая часть уравнения равна нулю, то знак равенства и нуль могут быть опущены.

> **solve(a*x^2+b*x+c,x);**

$$\frac{-b+\sqrt{b^2-4\,a\,c}}{2\,a}, \frac{-b-\sqrt{b^2-4\,a\,c}}{2\,a}$$

Если найдено несколько решений уравнения, то корни записываются в виде последовательности.

Аналогично может быть получено решение для неравенства.

> **solve(x^2+x>5,x);**

$$\text{RealRange}\left(-\infty, \text{Open}\left(-\frac{1}{2}-\frac{\sqrt{21}}{2}\right)\right), \text{RealRange}\left(\text{Open}\left(-\frac{1}{2}+\frac{\sqrt{21}}{2}\right), \infty\right)$$

Open – открытый диапазон, т.е. указанное в скобках значение в него не входит.

Если в качестве первого параметра функции **solve** будет множество, состоящее из уравнений, то *Maple* будет рассматривать это множество как систему.

Решим систему линейных уравнений

> **solve({x+5*y+z=1,2*x-y+4*z=4,x+2*y+2*z=12}, {x,y,z});**

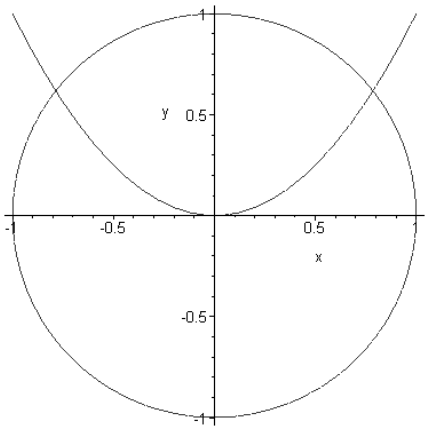
$$\{z = 23, x = -42, y = 4\}$$

Рассмотрим еще один пример. Решим систему нелинейных уравнений

$$\begin{cases} y = 2x^2 \\ x^2 + y^2 = 1 \end{cases},$$

предварительно проиллюстрировав решение графически.

> **plots[implicitplot]({y=x^2,x^2+y^2=1},x=-1..1, y=-1..1);**



> **solve({y=x^2,x^2+y^2=1},{x,y});**

$$\{x = \text{RootOf}(-\text{RootOf}(\_Z + \_Z^2 - 1, label = \_L1) + \_Z^2, label = \_L2),$$
$$y = \text{RootOf}(\_Z + \_Z^2 - 1, label = \_L1)\}$$

В решении присутствует выражение RootOf, означающее, что решение получено в неявной форме. Для вычисления решения в явной форме следует воспользоваться функцией **allvalues**.

> **allvalues(%);**

$$\{y = -\frac{1}{2} + \frac{\sqrt{5}}{2}, x = \frac{\sqrt{-2 + 2\sqrt{5}}}{2}\}, \{y = -\frac{1}{2} + \frac{\sqrt{5}}{2}, x = -\frac{\sqrt{-2 + 2\sqrt{5}}}{2}\},$$
$$\{y = -\frac{1}{2} - \frac{\sqrt{5}}{2}, x = \frac{\sqrt{-2 - 2\sqrt{5}}}{2}\}, \{y = -\frac{1}{2} - \frac{\sqrt{5}}{2}, x = -\frac{\sqrt{-2 - 2\sqrt{5}}}{2}\}$$

> **evalf(%);**

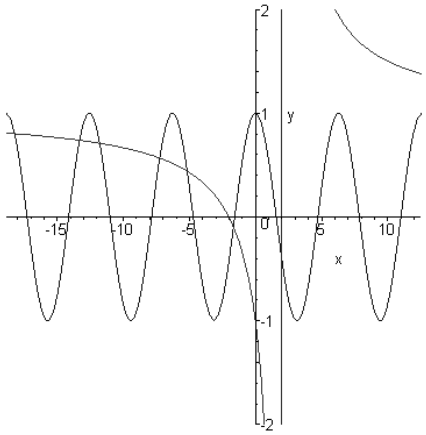
$\{y = 0.6180339880, x = 0.7861513775\}$, $\{y = 0.6180339880, x = -0.7861513775\}$, $\{y = -1.618033988, x = 1.272019650\ I\}$,
$\{y = -1.618033988, x = -1.272019650\ I\}$

После преобразования полученного решения к виду с плавающей точкой стало очевидным, что система имеет два действительных корня (их видно на графике) и два комплексных.

Если по каким-либо причинам с помощью функции **solve** не удалось найти решение, то можно использовать функцию **fsolve** для нахождения решения численным методом.

Решим уравнение $\cos(x) - \frac{x+2}{x-2} = 0$. Предварительно для выяснения количества корней построим графики функций $y = \cos(x)$ и $y = \frac{x+2}{x-2}$.

> **plot({cos(x),(x+2)/(x-2)}, x=-6*Pi..4*Pi,**
y=-2..2,color=[red, blue]);



Как видно из графика, гипербола $y = \frac{x+2}{x-2}$ имеет вертикальную асимптоту $x = 2$ и горизонтальную – $y = 1$. Таким образом, предложенное для решения уравнение имеет бесконечное количество корней в диапазоне от нуля до -∞. Пробуем решить уравнение при помощи функции **fsolve.**

> **fsolve(cos(x)-(x+2)/(x-2),x);**

-1.662944360

Найден ближайший к нулю корень. Для того, чтобы найти следующий корень функции **fsolve** необходимо указать интервал для поиска, при этом крайне желательно чтобы на этом интервале был только один корень. Итак, найдем второй корень.

> **fsolve(cos(x)-(x+2)/(x-2),x=-6..-4);**

-5.170382990

6 УПРАВЛЯЮЩИЕ КОНСТРУКЦИИ

Любой алгоритм можно реализовать, используя лишь три управляющие конструкции: последовательное выполнение, ветвление и цикл.

Оператор ветвление можно изобразить следующей синтаксической диаграммой:

if <условие> then <последовательность операторов>
[elif <условие> then < последовательность операторов >]
[else < последовательность операторов >]
end if

Выражение **elif** следует понимать как **else-if,** т.е. «в противном случае проверить следующее условие». В квадратных скобках – необязательные параметры.

Покажем, как можно вычислить корни квадратного уравнения с использованием оператора ветвления **if**:

```
> a:=2;b:=6;c:=1;
d:=b^2-4*a*c;
if d>0 then (-b+sqrt(d))/2/a,(-b-sqrt(d))/2/a
  elif d=0 then -b/2/a
  else print(`Действительных корней нет !!!`)
end if;
```

$$a := 2$$
$$b := 6$$
$$c := 1$$
$$d := 28$$
$$-\frac{3}{2}+\frac{\sqrt{7}}{2}, -\frac{3}{2}-\frac{\sqrt{7}}{2}$$

Оператор ветвления можно задавать в форме функции, при этом **if** должно быть заключено в обратные кавычки:

`if `(условие, **истинное выражение**, **ложное выражение**)

Например,

```
> d:=4;
   `IF`(D>=0, `ЕСТЬ ДЕЙСТВИТЕЛЬНЫЕ КОРНИ.`, `ДЕЙСТВИТЕЛЬНЫХ КОРНЕЙ НЕТ !!!`);
```

$d := 4$

Есть действительные корни.

```
> d:=-2;
`if`(d>=0, `Есть действительные корни.`, `Действительных корней нет !!!`);
```

$d := -2$

Действительных корней нет !!!

Циклы в *Maple* можно задавать двух типов: **for-to** и **while**.

[for <имя>] [from <выражение>] [by < выражение >]
[to < выражение >] [while < выражение >]
do < последовательность операторов > end do;

или

[for <имя>] [in < выражение >] [while < выражение >]
do < последовательность операторов > end do;

По умолчанию значения выражений **from** и **by** равны единице.

Приведем примеры использования циклов. В следующем примере организован цикл по переменной *k* от нуля до 3 с шагом 0,5.

```
> for k from 0 to 3 by 0.5 do print(k) end do;
```

0
0.5
1.0
1.5
2.0
2.5
3.0

Выше, при рассмотрении функции **fsolve**, были найдены два ближайших к нулю корня уравнения $\cos(x) - \frac{x+2}{x-2} = 0$. Принимая во внимание период функции cos(*x*) нетрудно заметить, что в каждый интервал $[-i\cdot\pi, -i\cdot\pi + \pi]$, где $i = 1, 2, 3, 4, \ldots, n$, попадает только один корень. Следующий пример демонстрирует нахождение *n* первых корней, указанного выше уравнения.

```
> n:=10;
```

$n := 10$

```
> for i from 1 to n do
fsolve(cos(x)-(x+2)/(x-2),x=-i*Pi..-i*Pi+Pi);
end do;
```

-1.662944360
-5.170382990
-7.250409918
-11.78482522
-13.30607789
-18.20951859
-19.46987598

-24.57696992
-25.67706817
-30.91781263

Следующие примеры демонстрирует вычисление *n*! с использованием цикла **for**.

```
> n:=7;
z:=1:
for x in seq(i,i=1..n) do z:=z*x;
 if x=n then print(z) end if
end do:
```

n := 7
5040

```
> n:=7;
z:=1:
for x while (x<=n) do z:=z*x:
 if x=n then print(z) end if
end do:
```

n := 7
5040

В циклах можно использовать операторы **next** и **break**. Оператор **next** используется для перехода к следующей итерации цикла, не завершив текущую. Оператор **break** используется для прерывания цикла. В следующем примере показано, как можно использовать операторы **next** и **break** для печати нечетных чисел в диапазоне от 1 до 10.

```
>  for i  do
 if (i > 10) then break end if;
 if (i mod 2)=0 then next end if;
 print(i)
 end do;
```

1
3
5
7
9

7 ПРОЦЕДУРЫ И ФУНКЦИИ

Процедуры являются важным элементом структурного программирования и служат средством расширения возможностей *Maple* пользователем. Процедуры имеют имя и список параметров, даже если он пустой. Процедуры вызываются, также как встроенные функции, указанием их имени со списком фактических параметров.

Общая форма задания процедуры:

proc (**формальные параметры**)
local локальные параметры;
global глобальные параметры;
options расширяющие ключи;
description комментарии;
тело процедуры
end proc

С помощью знака **::** после имени переменной можно определить ее тип. Несоответствие фактических параметров типу заданных переменных ведет к сообщению об ошибке и к отказу от выполнения процедуры.

При рассмотрении ветвления, было предложено решение квадратного уравнения. Оформим это решение в виде процедуры. Объявим фактические параметры алгебраическим типом, в который входят целые, дробные и числа с плавающей точкой.

```
> A:=proc(a::algebraic,b::algebraic,c::algebraic)
local d;
description "Решение квадратного уравнения";
d:=b^2-4*a*c;
 if d>0 then (-b+sqrt(d))/2/a,(-b-sqrt(d))/2/a
   elif d=0 then -b/2/a
   else print(`Действительных корней нет !!!`)
 end if;
end proc;
```

```
A := proc(a::algebraic, b::algebraic, c::algebraic)
local d;
description "Решение квадратного уравнения";
    d := b^2 - 4×a×c;
    if 0 < d then 1/2×(−b + sqrt(d))/a, 1/2×(−b − sqrt(d))/a
elif d = 0 then −1/2×b/a
else print(`Действительных корней нет !!!`)
end if
end proc
```

```
> A(1,12,3);
```

$$-6+\sqrt{33}, -6-\sqrt{33}$$

```
> A(1,8.5,3);
```

, -0.3689563260, -8.131043675

```
> A(12,4,2);
```

Действительных корней нет !!!

Если в качестве параметра будет подставлено значение недопустимого типа, это приведет к ошибке и сообщению о недопустимом типе параметра.

```
> A(12,"4",6);
```

Error, invalid input: A expects its 2nd argument, b, to be of type algebraic, but received 4

Если в теле процедуры имеются операции присвоения для ранее определенных (глобальных) переменных, то изменение их значений в ходе выполнения процедуры создает так называемый *побочный эффект*. Он способен существенно изменить алгоритм решения задач, и поэтому, как правило, недопустим. Встречая такие операции присвоения, *Maple* корректирует текст процедуры, добавляя в нее объявление переменных с помощью **local.** При этом выдается предупреждение вида:

Warning, `d` is implicitly declared local to procedure `A`

Если все-таки работа с глобальными переменными внутри процедуры необходима, то эти переменные должны быть объявлены в процедуре с помощью **global**.

Следует отметить, что нельзя делать глобальными переменные, указанные в списке параметров процедуры, поскольку они уже фактически объявлены локальными. Такая попытка приведет к появлению сообщения об ошибке.

Расширяющие ключи используются для детальной настройки процедуры.

Ключ *operator* определяет, что с процедурой можно работать как с оператором, а ключ *arrow* дополнительно показывает, что оператор записывается в стрелочной нотации. Покажем на примере:

```
> f:=proc(x) option operator, arrow; sin(x)+cos(x) end;
```

$$f := x \rightarrow \sin(x) + \cos(x)$$

Последняя процедура эквивалентна записи:

> **g:=x->sin(x)+cos(x);**

$$g := x \to \sin(x) + \cos(x)$$

Вычислим значения $f(x)$ и $g(x)$ в точке x=0,5:

> **f(0.5);**

1.357008100

> **g(0.5);**

1.357008100

Проверим эквивалентность $f(x)$ и $g(x)$:

> **evalb(f(x)=g(x));**

true

Такую форму процедуры иногда называют *процедура-функция*. Следующий пример демонстрирует процедуру-функцию с двумя параметрами.

> **z:=(x,y)->x^2+y^2;**

$$z := (x, y) \to x^2 + y^2$$

> **z(1.2,3.5);**

13.69

Процедура функция может не иметь параметров, но при ее вызове обязательно сохраняется пара скобок.

> **e:=()->evalf(exp(1));**

$$e := (\ ) \to \mathrm{evalf}(\mathbf{e})$$

> **e;**

e

> **e();**

2.718281828

> **ln(e());**

0.9999999998

> **ln(exp(1));**

1

Функция **unapply** используется для конструирования функций из выражений.

> **p := x^2 + sin(x) + 1;**

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

> **f := unapply(p,x);**

$$f := x \to x^2 + \sin(x) + 1$$

> **f(Pi/12);**

$$\frac{\pi^2}{144} + \sin\left(\frac{\pi}{12}\right) + 1$$

Для задания кусочной функции в *Maple* имеется специальное средство – функция **piecewise**, которая имеет следующие параметры:

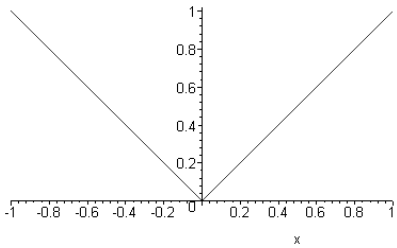
piecewise(интервал_1, **выражение_1**, **интервал_2**, **выражение_2**, ...,
интервал_*n*, **выражение_*n* [, выражение])**

Последнее необязательное выражение задает функцию для неохваченных интервалов.

> **p :=x-> piecewise(x<0,-x,x>0,x);**

$$p := x \to \mathrm{piecewise}(x < 0, -x, 0 < x, x)$$

> **plot(p(x),x=-1..1,scaling=constrained);**



Кусочные функции интегрируются и дифференцируются теме же средствами, что и обычные функции. В следующем примере для вычисления объема и площади поверхности тела, образованного вращением вокруг оси z кусочной функции $f(z)$, использовались известные выражения:

$$V = \pi \int_a^b (f(z))^2 dz$$

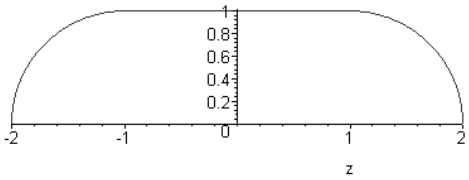
и

$$S = 2\pi \int_a^b f(z) \sqrt{1 + \left(\frac{df(z)}{dz} \right)^2} \, dz .$$

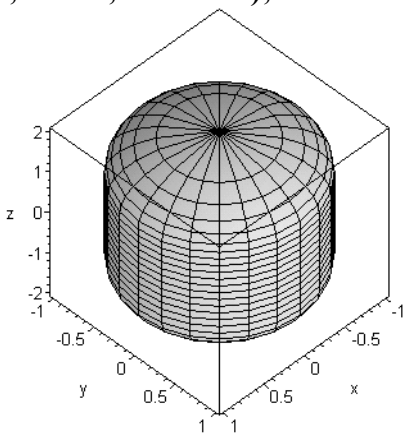
```
> f := z->piecewise( z>-2 and z<-1,sqrt(1-(z+1)^2),
        z>=-1 and z<=1,1,
        z>1 and z <2,sqrt(1-(z-1)^2));
```

$$f := z \to \text{piecewise}(-2 < z \text{ and } z < -1, \sqrt{1-(z+1)^2}, -1 \le z \text{ and } z \le 1, 1, 1 < z \text{ and } z < 2, \sqrt{1-(z-1)^2})$$

```
> plot(f(z),z=-2..2,scaling=constrained);
```



```
> plots[cylinderplot](f(z),theta=0..2*Pi,z=-2..2, axes=box);
```



```
> V:=Pi*int(f(z)^2,z=-2..2);
```

$$V := \frac{10\,\pi}{3}$$

```
> S:=2*Pi*int(f(z)*sqrt(1+diff(f(z),z)^2), z=-2..2);
```

$$S := 8\,\pi$$

8 РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

В тех случаях, когда научную или техническую проблему можно сформулировать математически, наиболее вероятно, что задача сведется к одному или нескольким дифференциальным уравнениям. Это всегда имеет место для широкого класса проблем, связанных с силами и движением.

В гидро- и аэродинамике, теплотехнике, радиотехнике и многих других областях науки и техники большое количество задач сводится к дифференциальным уравнениям. Однако, несмотря на большие

усилия, которые более двух столетий прилагают многие математики мира, число типов дифференциальных уравнений, разрешенных в замкнутом виде или в квадратурах, остается очень ограниченным. Поэтому в настоящее время существует множество проблем, точно сформулированных в виде дифференциальных уравнений, решение которых еще не найдены.

Все это привело к тому, что наряду с аналитическими и приближенными методами начали широко применяться численные методы решения дифференциальных уравнений, роль которых особенно возросла с применением ЭВМ.

Maple имеет средства для аналитического, приближенного и численного решения как обыкновенных дифференциальных уравнений (*ordinary differential equations – ODEs*), так и дифференциальных уравнений в частных производных (*partial differential equations – PDEs*), а также их систем.

8.1 РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Для решения обыкновенных дифференциальных уравнений используется функция:

dsolve(*ODE*, *y*(*x*), [параметры]),

где *ODE* – обыкновенное дифференциальное уравнение; *y* (*x*) – искомая функция.

Необязательные параметры задаются в форме *ключевое слово = значение*. *Ключевое слово* может быть опущено, и в этом случае параметром будет являться только *значение*. Функция **dsolve** позволяет найти решение многих дифференциальных уравнений. По умолчанию **dsolve** пытается найти точное (аналитическое) решение. Однако, если точное решение не может быть получено, то можно попытаться найти приближенное решение с помощью разложения в ряд (параметр *type = series*) или численным методом (параметр *type = numeric*).

Найдем общее решение дифференциального уравнения

$$\frac{\frac{d}{dx}y(x)}{2x} + 3xy(x) = e^{-2x^3}.$$

> **ODE:=diff(y(x),x)/(2*x)+3*x*y(x)=exp(-2*x^3);**

$$ODE := \frac{1}{2}\,\frac{\frac{d}{dx}\mathrm{y}(x)}{x} + 3\,x\,\mathrm{y}(x) = \mathbf{e}^{(-2x^3)}$$

> **dsolve(ODE);**

$$\mathrm{y}(x) = (x^2 + \_C1)\,\mathbf{e}^{(-2x^3)}$$

Найдено общее решение дифференциального уравнения. Отметим, что исходное дифференциальное уравнение содержало только одну функцию – *y* (*x*), относительно которой возможно решение, поэтому в функции **dsolve** второй параметр может быть опущен.

Найдем частное решение дифференциального уравнения при условии, что *y* (0) = 5. Можно определить постоянную интегрирования (*Maple* обозначил ее *_C1*), используя уже полученное решение дифференциального уравнения и дополнительное условие. Но если в качестве первого параметра в функции **dsolve** подставить множество или список из дифференциального уравнения и дополнительного условия, то *Maple* сразу найдет частное решение дифференциального уравнения.

> **R1:=dsolve({ODE,y(0)=5});**

$$R1 := y(x) = (x^2 + 5)\, e^{(-2x^3)}$$

Найдем решение рассмотренного выше дифференциального уравнения с помощью рядов. Предварительно установим максимальную степень ряда – 16, напомним, что по умолчанию это значение равно 6.

> **Order:=16:**
R2:=dsolve({ODE,y(0)=5},y(x),series);

$$R2 := y(x) = 5 + x^2 - 10x^3 - 2x^5 + 10x^6 + 2x^8 - \frac{20}{3}x^9 - \frac{4}{3}x^{11} +$$
$$+ \frac{10}{3}x^{12} + \frac{2}{3}x^{14} - \frac{4}{3}x^{15} + O(x^{16})$$

Используя полученное в виде ряда решение, организуем список из координат точек в диапазоне *x* от 0 до 1, для последующего вывода этих точек функцией **plot**.

> **R2p:=[seq([i/25,subs(x=i/25, op(2,convert(R2,polynom)))], i=0..25)]:**

Найдем решение дифференциального уравнения численным методом. *Maple* умеет решать дифференциальные уравнения различными численными методами. По умолчанию используется метод Рунге-Кутта четвертого-пятого порядка.

> **R3:=dsolve({ODE,y(0)=5},numeric);**

$$R3 := \mathbf{proc}\ (x\_rkf45)\ \ldots\ \mathbf{end\ proc}$$

При численном решении дифференциальных уравнений функции **dsolve** создает процедуру. При вызове процедуры, подставляя в качестве параметра значение аргумента, выводится список, состоящий из аргумента и соответствующего значения функции. В общем случае при численном решении дифференциальных уравнений *n*-го порядка также выводятся значения всех производных до *n*-1 порядка.

> **R3(0.12);**

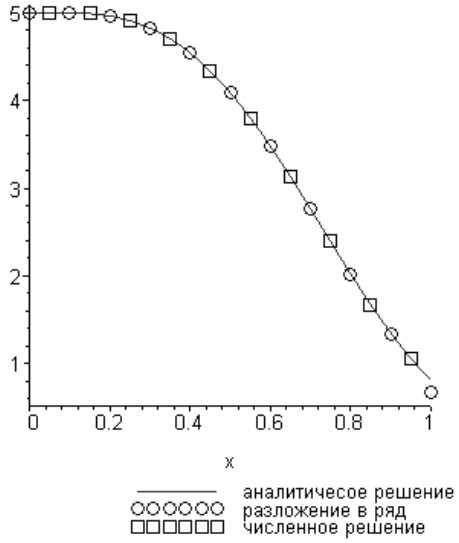
$$[x = 0.12,\ y(x] \qquad 709897276009496\ ]$$

Используя полученное в виде процедуры решение, организуем список из координат точек в диапазоне *x* от 0 до 1, для последующего вывода этих точек функцией **plot**.

> **R3p:=[seq([i/25+0.02,op(2,op(2,R3(i/25+0.02)))], i=0..25)]:**

Совместим на одной координатной плоскости аналитическое решение, решение при помощи ряда и численное решение.

> **plot([rhs(R1), R2p, R3p],x=0..1, style=[line,point,point], color=[red,blue,black],symbol=[box, circle], symbolsize=[17,17],legend=["аналитичесое решение", "разложение в ряд","численное решение"]);**



Обратим внимание на следующие параметры функции **plot**:

- *symbol* – способ отображения точек;
- *symbolsize* – размер точек (по умолчанию – 10);
- *legend* – подрисуночная надпись (легенда).

Для визуализации численных решений дифференциальных уравнений в пакете *plots* имеется функция **odeplot**. В предыдущем примере, для построения решения на интервале от 0 до 1, функцию **odeplot** можно использовать следующим образом

> **plots[odeplot](R3,0..1);**

Для решения систем дифференциальных уравнений первым параметром функции **dsolve** должно быть множество или список дифференциальных уравнений входящих в систему, а в случае нахождения частного решения туда же должны входить и дополнительные условия. Найдем решение системы дифференциальных уравнений

$$\begin{cases} \dfrac{dx(t)}{dt} = y(t) - x(t) \\ \dfrac{dy(t)}{dt} = -x(t) - 3y(t) \end{cases}$$

удовлетворяющее условию: $x(1) = 0$, $y(1) = 1$.

> **SYS:={diff(x(t),t)=y(t)-x(t),**
diff(y(t),t)=-x(t)-3*y(t),
x(1)=0,y(1)=1};

$$SYS := \{\frac{d}{dt}\mathrm{x}(t) = \mathrm{y}(t) - \mathrm{x}(t), \frac{d}{dt}\mathrm{y}(t) = -\mathrm{x}(t) - 3\,\mathrm{y}(t), \mathrm{x}(1) = 0, \mathrm{y}(1) = 1\}$$

Аналитическое решение

> **R1:=dsolve(SYS,{x(t),y(t)});**

$$R1 := \{\mathrm{y}(t) = -\mathbf{e}^{(-2t)}\left(-\frac{2}{\mathbf{e}^{(-2)}} + \frac{t}{\mathbf{e}^{(-2)}}\right), \mathrm{x}(t) = \mathbf{e}^{(-2t)}\left(-\frac{1}{\mathbf{e}^{(-2)}} + \frac{t}{\mathbf{e}^{(-2)}}\right)\}$$

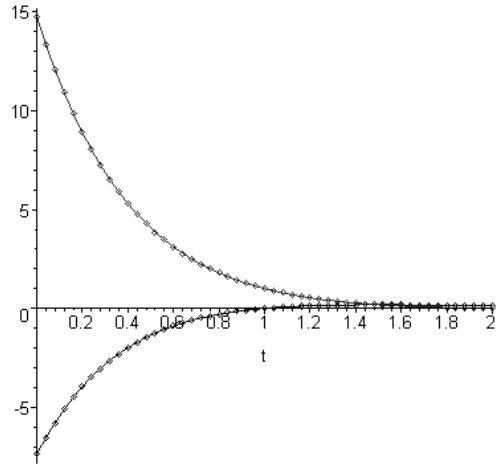
> **A:=plot([rhs(R1[1]),rhs(R1[2])],t=0..2, color=[blue,red]):**

Численное решение

> **R2:=dsolve(SYS,{x(t),y(t)},numeric);**

$$R2 := \mathbf{proc}\ (x\_rkf45)\ \ldots\ \mathbf{end\ proc}$$

> **B:=plots[odeplot](R2,[[t,x(t),color=blue, style=point],[t,y(t),color=red,style=point]],0..2):**
> **plots[display](A,B);**



На графике сплошными линиями показано аналитическое решение системы дифференциальных уравнений, а точками – численное.

При решении дифференциальных уравнений выше первого порядка бывает необходимо, в качестве дополнительных условий, задать не только значения функции в какой-либо точке, но и значения производных. Для этого используется дифференциальный оператор – **D**. Если, например, значение производной функции $y(t)$, в точке $t = 0$ равно 5, то необходимо записать – **D**(y)(0) = 5. Тоже для второй производной – **D** (**D**(y))(0) = 5 или (**D**@@2)(y)(0) = 5, для третьей – **D** (**D** (**D**(y)))(0) = 5 или (**D**@@3)(y)(0) = 5 и т.д. Таким образом, производную n-го порядка можно записать как (**D**@@n)(y)(0).

Решим дифференциальное уравнение $y''' + 2y' + 12y = 0$ при дополнительных условиях: $y(0) = 4$, $y'(0) = 0$, $y''(0) = 0$.

> **de:=diff(y(t),t$3)+2*diff(y(t),t)+12*y(t)=0;**

$$de := \left(\frac{d^3}{dt^3}\,\mathrm{y}(t)\right) + 2\left(\frac{d}{dt}\,\mathrm{y}(t)\right) + 12\,\mathrm{y}(t) = 0$$

> **init:=y(0)=4,D(y)(0)=0,(D@@2)(y)(0)=0;**

$$init := \mathrm{y}(0) = 4,\ \mathrm{D}(y)(0) = 0,\ (\mathrm{D}^{(2)})(y)(0) = 0$$

> **dsolve({de,init});**

$$\mathrm{y}(t) = \frac{12}{7}\,\mathbf{e}^{(-2t)} + \frac{8}{35}\sqrt{5}\,\mathbf{e}^{t}\sin(\sqrt{5}\,t) + \frac{16}{7}\,\mathbf{e}^{t}\cos(\sqrt{5}\,t)$$

8.2 РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

Для решения дифференциальных уравнений в частных производных используется функция:

pdsolve(*PDE*, $u(x, y)$, [параметры]),

где *PDE* – дифференциальное уравнение в частных производных; $u(x, y)$ – искомая функция нескольких переменных.

Необязательные параметры выполняют приблизительно туже роль, что и в функции **dsolve** и более подробно о них будет рассказано ниже.

Для решения системы дифференциальных уравнений в частных производных, в качестве первых двух параметров следует использовать множества или списки с уравнениями и искомыми функциями.

Типичной особенностью дифференциальных уравнений в частных производных и их систем является то, что для однозначного определения частного решения здесь требуется задание не значений того или иного конечного числа параметров, а некоторых функций. Найдем общее решением уравнения

$$\frac{\partial^2 u(x,t)}{\partial t^2} = \frac{\partial^2 u(x,t)}{\partial x^2}$$

> **PDE:=diff(u(x,t),t,t)= diff(u(x,t),x,x);**

$$PDE := \frac{\partial^2}{\partial t^2}\mathrm{u}(x,t) = \frac{\partial^2}{\partial x^2}\mathrm{u}(x,t)$$

> **pdsolve(PDE,u(x,t));**

$$\mathrm{u}(x,t) = \_\mathrm{F1}(t+x) + \_\mathrm{F2}(t-x)$$

Таким образом, рассмотренное дифференциальное уравнение лишь в той мере ограничивает произвол в выборе функций двух переменных *u*, что ее удается выразить через сумму двух функций *_F1* и *_F2* от одного переменного, которые остаются произвольными, если не дано каких-либо дополнительных условий.

Если требуется найти решение в виде произведения функций, то следует использовать параметр *HINT = `*`*.

> **pdsolve(PDE,u(x,t),HINT=`*`);**

$$\left(\mathrm{u}(x,t) = \_\mathrm{F1}(\_\xi 1)\ \_\mathrm{F2}(\_\xi 2)\right)\ \&\mathrm{where}$$

$$\left[\left\{\frac{d}{d\_\xi 2}\_\mathrm{F2}(\_\xi 2) = \_c_2,\ 2\left(\frac{d}{d\_\xi 1}\_\mathrm{F1}(\_\xi 1)\right)\_c_2 = 0\right\},\ \&\mathrm{and}\left(\left\{\_\xi 1 = t - x,\ \_\xi 2 = \frac{x}{2} + \frac{t}{2}\right\}\right)\right]$$

При использовании параметра *build – Maple* попытается записать решение в явном виде.

> **pdsolve(PDE,u(x,t),HINT=`*`, build);**

$$\mathrm{u}(x,t) = \frac{1}{2}\_C1\ \_c_2\, x + \frac{1}{2}\_C1\ \_c_2\, t + \_C1\ \_C2$$

где *_C1*, *_C2* и $\_c_2$ – постоянные интегрирования, которые определяются из так называемых краевых или начальных условий, позволяющих однозначно выделить интересующее решение. В то время как краевые условия задаются исключительно на граничных точках области, где ищется решение, начальные условия могут оказаться заданными на определенном множестве точек внутри области.

Пусть требуется найти функцию *F*(*x*, *y*), удовлетворяющую уравнению *Лапласа* $\frac{\partial^2 F(x,y)}{\partial x^2} + \frac{\partial^2 F(x,y)}{\partial y^2} = 0$, если известно, что на окружности $x^2 + y^2 = 16$ значения функции могут быть рассчитаны по выражению $x^2 y^2$.

Учитывая, что граничные условия заданы на окружности, целесообразно решать задачу в полярных координатах. Для преобразования дифференциального уравнения в полярную систему координат используем функцию **PDEchangecoords** из пакета *Detools***,** а для преобразования граничных условий – функция **dchange** из пакета PDEtools.

> **PDE:=diff(F(x,y),x,x)+diff(F(x,y),y,y);**

$$PDE := \left(\frac{\partial^2}{\partial x^2}\mathrm{F}(x,y)\right) + \left(\frac{\partial^2}{\partial y^2}\mathrm{F}(x,y)\right)$$

> **PDE:=DEtools[PDEchangecoords](PDE,[x,y],polar, [r,phi]);**

$$PDE := \frac{\left(\frac{\partial}{\partial r}\mathrm{F}(r,\phi)\right) r + \left(\frac{\partial^2}{\partial r^2}\mathrm{F}(r,\phi)\right) r^2 + \left(\frac{\partial^2}{\partial \phi^2}\mathrm{F}(r,\phi)\right)}{r^2}$$

> **dp:={x = r*cos(phi), y = r*sin(phi)};**

$$dp := \{x = r\cos(\phi), y = r\sin(\phi)\}$$

> **f:=x^2*y^2;**

$$f := x^2 y^2$$

> **f:=PDEtools[dchange](dp,f);**

$$f := r^4 \cos(\phi)^2 \sin(\phi)^2$$

Получаем решение в явном виде.

> **R:=rhs(pdsolve(PDE,HINT=`*`,build));**

$$R := \_C3 \sin(\sqrt{\_c_1}\ \phi)\ \_C1\ r^{(\sqrt{\_c_1})} + \frac{\_C3 \sin(\sqrt{\_c_1}\ \phi)\ \_C2}{r^{(\sqrt{\_c_1})}} +$$
$$+ \_C4 \cos(\sqrt{\_c_1}\ \phi)\ \_C1\ r^{(\sqrt{\_c_1})} + \frac{\_C4 \cos(\sqrt{\_c_1}\ \phi)\ \_C2}{r^{(\sqrt{\_c_1})}}$$

Будем считать, что искомая функция определена во всех точках действительной плоскости, включая центр координат, т.е. при $r = 0$. Следовательно, $\_C2 = 0$, так как в противном случае знаменатель в двух последних слагаемых решения обращается в ноль.

> **_C2:=0;**

$$\_C2 := 0$$

> **R;**

$$\_C3 \sin(\sqrt{\_c_1}\ \phi)\ \_C1\ r^{(\sqrt{\_c_1})} + \_C4 \cos(\sqrt{\_c_1}\ \phi)\ \_C1\ r^{(\sqrt{\_c_1})}$$

Первое и второе слагаемое содержат произведение двух констант, поэтому можно считать, что $\_C1$=1 и искать только $\_C3$ и $\_C4$.

> **_C1:=1;**

$$\_C1 := 1$$

> **R;**

$$\_C3 \sin(\sqrt{\_c_1}\ \phi)\ r^{(\sqrt{\_c_1})} + \_C4 \cos(\sqrt{\_c_1}\ \phi)\ r^{(\sqrt{\_c_1})}$$

Для нахождения констант $\_C3$ и $\_C4$ используем граничные условия.

> **simplify(subs({phi=0,r=4},R)) = eval(subs({phi=0,r=4},f)):**
> **simplify(subs({phi=Pi,r=4},R)) = eval(subs({phi=Pi,r=4},f)):**
> **solve({%,%%},{_C3,_C4});**

$$\{\_C3 = 0, \_C4 = 0\}$$

Очевидно, что найденное решение не удовлетворяет граничным условиям. Также отметим, функция **pdsolve** находит решение с точностью до некоторой постоянной. Поэтому добавим к полученному ранее решению постоянную Z, подлежащую определению.

> **R:=R+Z;**

$$R := \_C3 \sin(\sqrt{\_c_1}\ \phi)\ r^{(\sqrt{\_c_1})} + \_C4 \cos(\sqrt{\_c_1}\ \phi)\ r^{(\sqrt{\_c_1})} + Z$$

Используем граничные условия нахождения констант $\_C3$, $\_C4$ и Z, а также константы $\_c_2$.

> **R4:=subs(r=4,R=f);**

$$R4 := \_C3 \sin(\sqrt{\_c_1}\ \phi)\ 4^{(\sqrt{\_c_1})} + \_C4 \cos(\sqrt{\_c_1}\ \phi)\ 4^{(\sqrt{\_c_1})} + Z = 256\cos(\phi)^2 \sin(\phi)^2$$

> **simplify(subs(phi=0,R4)): U1:=simplify(%):**
> **simplify(subs(phi=Pi/2/_c[1]^(1/2),R4)): U2:=simplify(%):**

> **simplify(subs(phi=Pi/4,R4)): U3:=simplify(%):**
> **S:=solve({U1,U2,U3},{_C3,_C4,Z}):**
> **R4:=simplify(subs(S,R4)):**
> **_c[1]=solve(subs(phi=Pi,R4),_c[1]);**

$$_c_1 = (4,\ 16)$$

Константа $_c_1$ может принимать два значения 4 и 16. Подставляем в общее решение найденные значения констант *_C3*, *_C4* и *Z*.

> **R:=subs(S,R):**

Подстановка $_c_1 = 4$ приводит к ошибке – деление на ноль. Используем подстановку $_c_1 = 16$.

> **SOL:=combine(subs(_c[1]=16,R));**

$$SOL := 32 - \frac{1}{8}\cos(4\,\phi)\,r^4$$

Получено решение задачи в полярной системе координат. Нетрудно убедиться, что найденное решение удовлетворяет как уравнению *Лапласа*, так и граничным условиям. Преобразуем решение в декартову систему координат, используя известную связь между полярной и декартовой системой, а также приведенные ниже тригонометрические тождества.

> **pd:={r=sqrt(x^2+y^2),phi=arctan(y/x)};**

$$pd := \{\phi = \arctan\left(\frac{y}{x}\right),\ r = \sqrt{x^2 + y^2}\}$$

> **SOL:=PDEtools[dchange](pd,SOL);**

$$SOL := 32 - \frac{1}{8}\cos\left(4\arctan\left(\frac{y}{x}\right)\right)(x^2 + y^2)^2$$

> **cos(4*arctan(y/x)) = op(3,trigsubs(cos(4*arctan(y/x))));**

$$\cos\left(4\arctan\left(\frac{y}{x}\right)\right) = 2\cos\left(2\arctan\left(\frac{y}{x}\right)\right)^2 - 1$$

> **SOL:=subs(%,SOL);**

$$SOL := 32 - \frac{1}{8}\left(2\cos\left(2\arctan\left(\frac{y}{x}\right)\right)^2 - 1\right)(x^2 + y^2)^2$$

> **cos(2*arctan(y/x))^2 = op(11,trigsubs(cos(2*arctan(y/x))^2));**

$$\cos\left(2\arctan\left(\frac{y}{x}\right)\right)^2 = \frac{\left(1 - \tan\left(\arctan\left(\frac{y}{x}\right)\right)^2\right)^2}{\left(1 + \tan\left(\arctan\left(\frac{y}{x}\right)\right)^2\right)^2}$$

> **SOL:=simplify(subs(%,SOL));**

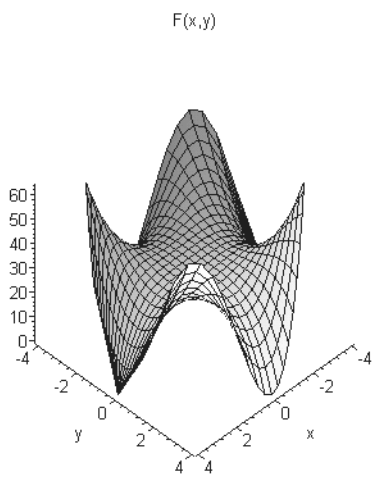
$$SOL := 32 - \frac{1}{8}x^4 + \frac{3}{4}x^2y^2 - \frac{1}{8}y^4$$

> **F:=unapply(SOL,x,y);**

$$F := (x, y) \to 32 - \frac{1}{8}x^4 + \frac{3}{4}x^2y^2 - \frac{1}{8}y^4$$

Построим график функции $F(x,y)$ в круге радиусом 4.

> **plot3d(F(x,y), x=-4..4, y=-sqrt(16-x^2)..sqrt(16-x^2), axes=framed, title="F(x,y)");**



8.3 ЧИСЛЕННОЕ РЕШЕНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ

Рассмотрим следующую задачу. Пусть требуется определить процесс поперечных колебаний струны круглого поперечного сечения радиусом $r = 0{,}002$ м и длиной $L = 1$ м с жестко закрепленными концами. В начальный момент времени струна имела форму квадратичной параболы, симметричной относительно перпендикуляра к середине струны и максимальным отклонением $h = 0{,}01$ м, а затем отпущена без толчка. Натяжение струны $T = 100$ Н, а плотность материала струны $\rho_c = 7800$ кг/м^3.

Поперечные колебания струны описываются *волновым уравнением*

$$\frac{\partial^2 u(x,t)}{\partial t^2} = a^2 \frac{\partial^2 u(x,t)}{\partial x^2},$$

где $a = \sqrt{\frac{T}{\rho_l}}$, $\rho_l = \rho S$ – линейная плотность струны, кг/м; $S = \pi r^2$ – площадь поперечного сечения струны.

Требуется найти решение при следующих *краевых условиях*:

раничные условия	**Начальные условия**
$u\,(0,\ t) = 0$	$u\,(x,\ 0) = f\,(x)$ $f\,(x)$ – форма струны в момент времени $t = 0$.
$u\,(l,\ t) = 0$	$\left.\frac{\partial u}{\partial t}\right\|_{t=0} = 0$

Запишем *волновое уравнение*:

> **PDE:=diff(u(x,t),t,t)=a^2*diff(u(x,t),x,x);**

$$PDE := \frac{\partial^2}{\partial t^2}\,\mathrm{u}(x,t) = a^2\left(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x,t)\right)$$

> **a:=sqrt(T/rho[l]);rho[l]:=rho[c]*S;S:=Pi*r^2;**

$$a := \sqrt{\frac{T}{\rho_l}}$$
$$\rho_l := \rho_c S$$
$$S := \pi\, r^2$$

> **PDE;**

$$\frac{\partial^2}{\partial t^2}\,\mathrm{u}(x,t) = \frac{T\left(\frac{\partial^2}{\partial x^2}\,\mathrm{u}(x,t)\right)}{\rho_c\,\pi\, r^2}$$

Введем исходные данные:

> **T:=100;L:=1;r:=0.002;rho[c]:=7800;h:=0.01;**

$$T := 100$$
$$L := 1$$
$$r := 0.002$$
$$\rho_c := 7800$$
$$h := 0.01$$

Запишем уравнение квадратичной параболы:

> **f:=A*x^2+B*x+C;**

$$f := A\,x^2 + B\,x + C$$

Найдем коэффициенты *A*, *B* и *C*:

> **solve({subs(x=0,A*x^2+B*x+C=0),**
subs(x=L/2,A*x^2+B*x+C=h),
subs(x=L,A*x^2+B*x+C=0)},
{A,B,C});

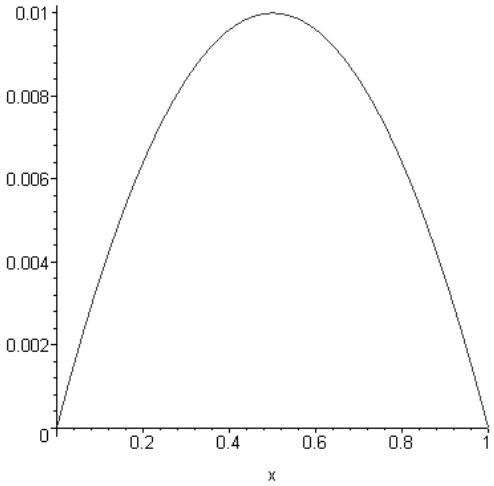
$$\{C = 0.,\ A = -0.40000000\ ,\ B = 0.40000000\ \}$$

Окончательно получим форму струны в начальный момент времени:

> **f:=subs(%,f);**

$$f := -0.04000000000x^2 + 0.04000000000x$$

> **plot(f,x=0..L);**



Запишем *краевые условия*:

> **BC:={u(0,t)=0,u(L,t)=0,u(x,0)=f,D[2](u)(x,0)=0};**

$$BC := \{u(0, t) = 0, D_2(u)(x, 0) = 0, u(1, t) = 0,\quad u(x, 0) = -0.04000000000\,x^2 + 0.04000000000\,x\}$$

Получим численное решение *волнового уравнения* с учетом краевых условий, задав величину шага по времени и координате:

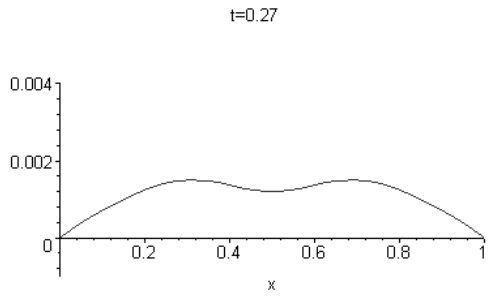
> **SOL:=pdsolve(PDE,BC,numeric,timestep=1/200, spacestep=1/200);**

SOL := **module** () **export** *plot* , *plot3d* , *animate* , *value* , *settings* ; ... **end module**

Полученный в результате модуль *SOL* позволяет визуализировать решение задачи, а также получить числовые значения перемещений точек в любой момент времени.

Построим положение струны в момент времени *t* = 0,27 с.

> **SOL:-plot(t=0.27,title="t=0.27");**



Создадим процедуру для расчета перемещений точек струны:

> **XT:=SOL:-value();**

$$XT := \textbf{proc}\ ()\ \ldots\ \textbf{end proc}$$

Получим перемещение середины струны в момент времени t = 0,27 с.

> **XT(L/2,0.27);**

$$[\,x = 0.50000000000000,\ t = 0.27,\ \mathrm{u}(x,t) = 0.00121214618006418979\,]$$

9 ПРИМЕРЫ РЕШЕНИЯ ИНЖЕНЕРНЫХ ЗАДАЧ

9.1 РАСЧЕТ БЫСТРО ВРАЩАЮЩИХСЯ ДИСКОВ

Задача о расчете быстро вращающихся дисков при постоянных свойствах материала E (модуль упругости), μ (коэффициент Пуассона) и переменной толщине $h(r)$ описывается системой уравнений:

$$\sigma_r(r)h(r) + r\left(\frac{d}{dr}\sigma_r(r)\right)h(r) + r\sigma_r(r)\left(\frac{d}{dr}h(r)\right) - \sigma_t(r)h(r) + \rho\omega^2 r^2 h(r) = 0, \quad (1)$$

$$\varepsilon_r(r) = \frac{d}{dr}u(r), \quad (2)$$

$$\varepsilon_t(r) = \frac{u(r)}{r}, \quad (3)$$

$$\varepsilon_r(r) = \frac{\sigma_r(r) - \mu\sigma_t(r)}{E} + \theta(r), \quad (4)$$

$$\varepsilon_t(r) = \frac{\sigma_t(r) - \mu\sigma_r(r)}{E} + \theta(r), \quad (5)$$

где $\sigma_r(r)$, $\sigma_t(r)$ – радиальные и окружные напряжения; $\varepsilon_r(r)$, $\varepsilon_t(r)$ – радиальные и окружные деформации; u (r) – радиальное перемещение, $\theta(r)$ – температурные деформации; ρ – плотность материала диска; ω – угловая скорость вращения.

> **restart;**

Запишем первое уравнение:

> **u1:=diff(R*sigma[r](R)*h(R),R)-sigma[t](R)*h(R)+rho*omega^2*R^2*h(R)=0;**

$$u1 := \sigma_r(R)\,\mathrm{h}(R) + R\left(\frac{d}{dR}\sigma_r(R)\right)\mathrm{h}(R) + R\,\sigma_r(R)\left(\frac{d}{dR}\mathrm{h}(R)\right) - \sigma_t(R)\,\mathrm{h}(R) + \rho\,\omega^2\,R^2\,\mathrm{h}(R) = 0$$

Продифференцировав третье уравнение в виде $u(R) = R\varepsilon_t(R)$ по R, и используя второе уравнение, можно записать:

> **u2:=epsilon[r](R)=epsilon[t](R)+R* diff(epsilon[t](R),R);**

$$u2 := \varepsilon_r(R) = \varepsilon_t(R) + R\left(\frac{d}{dR}\varepsilon_t(R)\right).$$

Подставив выражения для ε из четвертого и пятого уравнений в полученную зависимость получим:

> **u2:=simplify(subs({epsilon[r](R)=(1/E)* (sigma[r](R)-mu*sigma[t](R))+theta(R), epsilon[t](R)=(1/E)*(sigma[t](R)-mu*sigma[r](R))+theta(R)},u2));**

$$u2 := \frac{\sigma_r(R) - \mu\,\sigma_t(R) + \theta(R)\,E}{E} =$$
$$\frac{\sigma_t(R) - \mu\,\sigma_r(R) + \theta(R)\,E + R\left(\frac{d}{dR}\sigma_t(R)\right) - R\,\mu\left(\frac{d}{dR}\sigma_r(R)\right) + R\left(\frac{d}{dR}\theta(R)\right)E}{E}$$

Упростим выражение, разделив члены с $\sigma_t(R)$ и $\sigma_r(R)$:

> **u2:=isolate(u2,sigma[t](R));**

$$u2 := \mu\,\sigma_t(R) + \sigma_t(R) + R\left(\frac{d}{dR}\sigma_t(R)\right) =$$
$$\sigma_r(R) + \mu\,\sigma_r(R) + R\,\mu\left(\frac{d}{dR}\sigma_r(R)\right) - R\left(\frac{d}{dR}\theta(R)\right)E$$

Выразим $\sigma_t(R)$ из первого уравнения:

> **sigma[t](R):=solve(u1,sigma[t](R));**

$$\sigma_t(R) :=$$
$$\frac{\sigma_r(R)\,h(R) + R\left(\frac{d}{dR}\sigma_r(R)\right)h(R) + R\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right) + \rho\,\omega^2\,R^2\,h(R)}{h(R)}$$

Подставим полученное выражение для $\sigma_t(R)$ в уравнение u2:

> **u2:=simplify(u2);**

$$u2 := \left(\mu\,h(R)^2\,\sigma_r(R) + \mu\,h(R)^2\,R\left(\frac{d}{dR}\sigma_r(R)\right) + \mu\,h(R)\,R\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right)\right.$$
$$+ \mu\,h(R)^2\,\rho\,\omega^2\,R^2 + \sigma_r(R)\,h(R)^2 + 3\,R\left(\frac{d}{dR}\sigma_r(R)\right)h(R)^2$$
$$+ 2\,h(R)\,R\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right) + 3\,\rho\,\omega^2\,R^2\,h(R)^2 + R^2\left(\frac{d^2}{dR^2}\sigma_r(R)\right)h(R)^2$$
$$+ h(R)\,R^2\left(\frac{d}{dR}\sigma_r(R)\right)\left(\frac{d}{dR}h(R)\right) + h(R)\,R^2\,\sigma_r(R)\left(\frac{d^2}{dR^2}h(R)\right)$$
$$\left. - R^2\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right)^2\right) \Big/ h(R)^2 =$$
$$\sigma_r(R) + \mu\,\sigma_r(R) + R\,\mu\left(\frac{d}{dR}\sigma_r(R)\right) - R\left(\frac{d}{dR}\theta(R)\right)E$$

Упростим полученное дифференциальное уравнение, собрав в правой части все члены с $\sigma_r(R)$:

> **u2:=isolate(u2,sigma[r](R));**

$$u2 := \mu\,h(R)\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right) + 3\left(\frac{d}{dR}\sigma_r(R)\right)h(R)^2 \qquad + h(R)\,R\left(\frac{d}{dR}\sigma_r(R)\right)\left(\frac{d}{dR}h(R)\right) + h(R)\,R\,\sigma_r(R)\left(\frac{d^2}{dR^2}h(R)\right)$$
$$+ 2\,h(R)\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right) + R\left(\frac{d^2}{dR^2}\sigma_r(R)\right)h(R)^2 \qquad - R\,\sigma_r(R)\left(\frac{d}{dR}h(R)\right)^2 =$$
$$-\left(\frac{d}{dR}\theta(R)\right)E\,h(R)^2 - \mu\,h(R)^2\,\rho\,\omega^2\,R - 3\,\rho\,\omega^2\,R\,h(R)^2$$

Таким образом, мы свели задачу, описываемую системой из пяти уравнений к решению дифференциального уравнения второго порядка относительно $\sigma_r(R)$. Окружные напряжения можно определить из выражения sigma[*t*](*R*), полученного ранее, а деформации и перемещение из исходных связей.

Рассмотрим, например, случай, когда неравномерность нагрева отсутствует, т.е. $\theta(R)$ = const:

> **theta(R):=theta;**

$$\theta(R) := \theta$$

> **u2:=simplify(u2);**

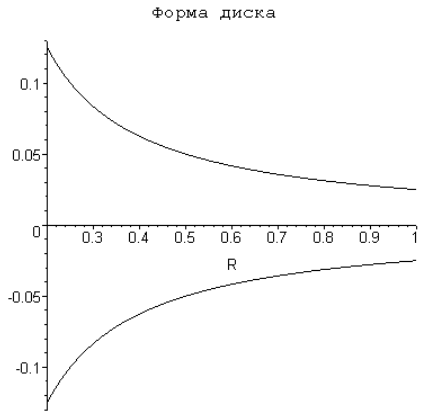
$$u2 := \mu\, h(R)\, \sigma_r(R)\left(\frac{d}{dR} h(R)\right) + 3\left(\frac{d}{dR}\sigma_r(R)\right) h(R)^2 + h(R)\, R\left(\frac{d}{dR}\sigma_r(R)\right)\left(\frac{d}{dR} h(R)\right) + h(R)\, R\, \sigma_r(R)\left(\frac{d^2}{dR^2} h(R)\right)$$
$$+ 2\, h(R)\, \sigma_r(R)\left(\frac{d}{dR} h(R)\right) + R\left(\frac{d^2}{dR^2}\sigma_r(R)\right) h(R)^2 - R\, \sigma_r(R)\left(\frac{d}{dR} h(R)\right)^2 = -\mu\, h(R)^2\, \rho\, \omega^2\, R - 3\, \rho\, \omega^2\, R\, h(R)^2$$

Зададим исходные данные и закон изменения толщины диска от радиуса, при этом радиус внутреннего отверстия примем равным 0,2 м, а наружный радиус диска 1 м:

> **mu:=1/3; rho:=7800; omega:=100; h(R):=1/20/R;**

$$\mu := \frac{1}{3}$$
$$\rho := 7800$$
$$\omega := 100$$
$$h(R) := \frac{1}{20\, R}$$

> **plot({h(R)/2,-h(R)/2},R=1/5..1,color=black, titlefont=[COURIER,12],title="Форма диска");**



Условия на границах диска:

> **ini:=sigma[r](1/5)=0,sigma[r](1)=0;**

$$ini := \sigma_r\left(\frac{1}{5}\right) = 0,\ \sigma_r(1) = 0$$

Решим дифференциальное уравнение u2 с граничными условиями ini:

> **u3:=evalf(simplify(dsolve({u2,ini},sigma[r](R))));**

$$u3 := \sigma_r(R) = 0.5656797953\ 10^8\ R^{0.7583057390} - \frac{853693.8027}{R^{1.758305739}} - 0.5571428571\ 10^8\ R^2$$

Найдем производную по R от полученного решения:

> **u4:=diff(rhs(u3),R);**

$$u4 := \frac{0.428958235210^8}{R^{0.2416942610}} + \frac{0.150105471310^7}{R^{2.758305739}} - 0.11142857141 0^9\ R$$

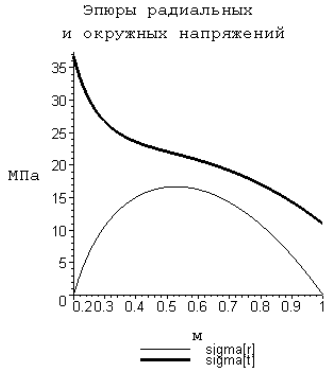
Подставим найденное решение относительно $\sigma_r(R)$ (переменная u3) и производную от него (переменная u4) в выражение для $\sigma_t(R)$:

> **u5:=eval(subs({diff(sigma[r](R),R)=u4, sigma[r](R)=rhs(u3)},sigma[t](R)));**

$$u5 := 20\left(\frac{0.214479117610^7}{R^{0.2416942610}} + \frac{75052.73565}{R^{2.758305739}} - 0.167142857010^7\ R\right) R$$

Построим эпюры радиальных и окружных напряжений в диске:

> **plot([rhs(u3)/1E6,u5/1E6],R=1/5..1,**
color=black,labelfont=[COURIER,12],labels=["м","МПа"],titlefont=[COURIER,12],title="Эпюры радиальных\n и окружных напряжений", legend=["sigma[r]","sigma[t]"], thickness=[1,3]);



9.2 РАСЧЕТ АППАРАТА ВЫСОКОГО ДАВЛЕНИЯ

Необходимо определить наружный радиус аппарата, толщину стенки, значение предельного давления, давление опрессовки, а также построить эпюры напряжений при рабочем давлении до и после автофретирования.

Исходные данные для расчета

внутренний радиус аппарата	0,6 м
рабочее давление	70 МПа
предел текучести материала	780 МПа
коэффициент запаса прочности по предельному состоянию	1,4
глубина пластических деформаций при автофретировании	40 %
условие пластичности	Губера-Мизеса
материал при деформации упрочняется	
модуль продольной упругости	$2{,}1 \cdot 10^5$ МПа
модуль упрочнения	5880 МПа

```
> restart:Digits:=8;
```

$$Digits := 8$$

```
> Sigma:=proc(x::string,P)
local L,Zn:
option operator,arrow:
description " Поцедура расчета напряжений ":
if (x="z" and nargs=1) then return ((Sigma("r")+Sigma("k"))/2):
end if:
if (x="z" and nargs=2) then return
(P*R1^2/(R2^2-R1^2))
:end if:
if x="r" then Zn:=-1: end if:
if x="k" then Zn:=+1: end if:
if nargs=1
   then
    L:=`if`(r>R[t],0,lambda): A*sigma[tau]/2 * (2*L*ln(r/R[t])+R[t]^2/R2^2+Zn*(1-L) *
R[t]^2/r^2+Zn*L):
   else
    P*R1^2/(R2^2-R1^2)*(1+Zn*R2^2/r^2):
```

end if:
end proc:

Ввод исходных данных

> **R1:=0.6;P[rab]:=70; sigma[tau]:=780; n[pred]:=1.4; T:=0.4;E[pu]:=210000; E[u]:=5880; A:=evalf(2/sqrt(3));**

$$R1 := 0.6$$
$$P_{rab} := 70$$
$$\sigma_{\tau} := 780$$
$$n_{pred} := 1.4$$
$$T := 0.4$$
$$E_{pu} := 210000$$
$$E_{u} := 5880$$
$$A := 1.1547005$$

Расчет параметра упрочнения

> **lambda:=1-E[u]/E[pu];**

$$\lambda := \frac{243}{250}$$

Расчет предельного давления

> **P[pred]:=A*n[pred]*P[rab];**

$$P_{pred} := 113.16065$$

Расчет наружного радиуса

> **R2:=solve(P[pred]=A*sigma[tau]/2 * (2*lambda*ln(R2/R1)+(1-lambda)*(R2^2/R1^2-1)),R2);**

$$R2 := 0.68000014$$

Расчет давления опрессовки

> **P[opres]:=A*sigma[tau]/2*(2*lambda*ln(R[t]/R1)-R[t]^2/R2^2+(1-lambda)*R[t]^2/R1^2+lambda);**

$$P_{opres} := 108.20162$$

Толщина стенки аппарата

> **Delta:=R2-R1;**

$$\Delta := 0.08000014$$

Расчет радиуса пластических деформаций

> **R[t]:=R1+Delta*T;**

$$R_{t} := 0.63200006$$

Расчет напряжений при рабочем и пробном давлении

> **sigma[r_rab]:=Sigma("r",P[rab]):**
sigma[k_rab]:=Sigma("k",P[rab]):
sigma[z_rab]:=Sigma("z",P[rab]):
> **sigma[r_prob]:= Sigma("r"):**
sigma[k_prob]:= Sigma("k"):
sigma[z_prob]:= Sigma("z"):

Расчет напряжений разгрузки и остаточных напряжений

> **sigma[r_raz]:=Sigma("r",P[opres]):**
sigma[k_raz]:=Sigma("k",P[opres]):
sigma[z_raz]:=Sigma("z",P[opres]):

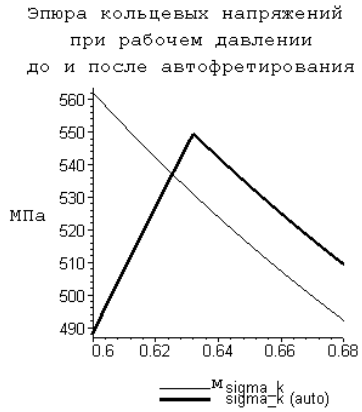
> **sigma[r_ost]:=sigma[r_prob]-sigma[r_raz]:**
sigma[k_ost]:=sigma[k_prob]-sigma[k_raz]:
sigma[z_ost]:=sigma[z_prob]-sigma[z_raz]:

Расчет напряжений при рабочем давлении после предварительного автофретирования

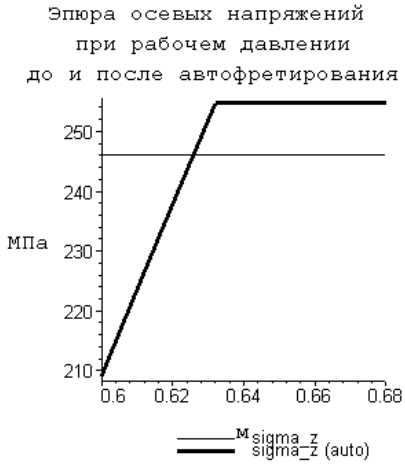
> **sigma[r_auto]:=sigma[r_rab]+sigma[r_ost]:**
sigma[k_auto]:=sigma[k_rab]+sigma[k_ost]:
sigma[z_auto]:=sigma[z_rab]+sigma[z_ost]:

Строим эпюры напряжений

> **plot([sigma[k_rab],sigma[k_auto]],r=R1..R2, color=[black,black,black,blue,blue,blue], labelfont=[COURIER,12],labels=["м","МПа"], titlefont=[COURIER,12],title="Эпюра кольцевых напряжений\n при рабочем давлении\n до и после автофретирования", legend=["sigma_к","sigma_к (auto)"], thickness=[1,3]);**



> **plot([sigma[z_rab],sigma[z_auto]],r=R1..R2, color=[black,black,black,blue,blue,blue], labelfont=[COURIER,12],labels=["м","МПа"],titlefont=[COURIER,12],title="Эпюра осевых напряжений\n при рабочем давлении\n до и после автофретирования", legend=["sigma_z","sigma_z (auto)"], thickness=[1,3]);**



Эпюра радиальных напряжений не приведена, вследствие их несущественных изменений.

Обратите внимание как процесс автофретирования приводит к перераспределению кольцевых и осевых напряжений.

СПИСОК ЛИТЕРАТУРЫ

1. Дьяконов В. П. Maple 7: Учебный курс. СПб.: Питер, 2002. 672 с.

2. Матросов А. В. Maple 6. Решение задач высшей математики и механики. СПб.: BHV, 2001. 528 с.

3. Аладьев В. З., Богдявичюс М. А. Maple 6: Решение математических, статистических и инженерно-физических задач. М.: Лаборатория базовых знании, 2001. 824 с.

4. Прочность. Устойчивость. Колебания. Справочник / Под ред. И. А. Биргера. М.: Машиностроение, 1968. Т. 2. 464 с.

5. Тимошенко С. П. Сопротивление материалов. М.: Наука, 1965. Т. 2. 480 с.

6. Гжиров Р. И. Краткий справочник конструктора. Л.: Машиностроение, 1983. 465 с.

7. Малинин Н. Н. Прикладная теория пластичности и ползучести. М.: Машиностроение, 1975. 400 с.

СОДЕРЖАНИЕ