

**Министерство образования и науки Украины
Севастопольский национальный технический
университет**

ВВЕДЕНИЕ В MAPLE

Методические указания
к выполнению лабораторной работы
для студентов, обучающихся по направлению
**6.050101 “Информационные управляющие
системы и технологии”**
очной и заочной форм обучения

**Севастополь
2010**

УДК 519.2

Введение в Maple: методические указания / Сост. Ю.В.Коваленко,
Е.Н. Заикина, С. А. Кузнецов. – Севастополь: Изд-во СевНТУ, 2010. – 28с.

Целью методических указаний является:

получение общего представления о математическом пакете MAPLE - одного из наиболее популярных представителей семейства систем автоматизации решений научно-технических задач. Изучение особенностей интерфейса, функциональных основных возможностей, формирования навыков практической работы в среде MAPLE, математических вычислений, моделирования, разработки приложений и анализа данных.

Методические указания составлены в соответствии с требованиями программы дисциплины «Теоретические основы информационных процессов» для студентов по направлению 6.050101 и утверждены на заседании кафедры информационных систем, протокол № 1 от 29 августа 2010 года.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент: Скороход Б.А., доктор технических наук, проф. кафедры ТК.

СОДЕРЖАНИЕ

Стр.

1. Цель работы	4
2. Что такое Maple?	4
3. Интерфейс Maple	4
4. Синтаксис языка Maple	5
4.1 Символы и переменные	5
4.2 Константы и внутренние функции	6
4.3 Типы данных	7
5. Основные математические операции	12
5.1 Операции с формулами	12
5.2 Операции с полиномами	15
5.3 Ограничения на переменные	16
6. Некоторые примеры математического анализа	16
7. Интегральные преобразования	20
8. Графика в Maple	21
9. Обзор библиотек Maple	26
10. Ход работы	27
11. Содержание отчёта	27
12. Контрольные вопросы	27
Список литературы	27

1. ЦЕЛЬ РАБОТЫ

Получение общего представления о математическом пакете MAPLE - одного из наиболее популярных представителей семейства систем автоматизации решений научно-технических задач. Изучение особенностей интерфейса, функциональных основных возможностей, формирования навыков практической работы в среде MAPLE, математических вычислений, моделирования, разработки приложений и анализа данных.

2. ЧТО ТАКОЕ MAPLE?

Maple – это программный пакет для автоматизации символьных и численных вычислений. Это означает, что вычисления в пакете можно проводить двумя способами: в символьном (аналитическом) виде и численными методами. В первом случае достигается наибольшая точность, но, к сожалению, как показывает практика, многие классы задач просто не возможно решить таким образом. И здесь приходят на помощь численные методы, огромное количество которых находятся во встроенных библиотеках.

Для написания программы на языке Maple не требуется глубоких знаний языков программирования. Все действия доступны любому пользователю, знакомому с Windows. Вы лишь концентрируете своё внимание на теоретической стороне решаемой задачи, а все действия, например, по решению линейных и не линейных алгебраических уравнений и систем, любого вида дифференциальных выражений, статистическому анализу, интегральным преобразованиям, выполнит за вас Maple.

В Maple включён редактор гипертекстовых документов. В одной среде пользователь решает свои задачи, оформляет и распечатывает документы высокого качества и сложности. Достаточно отметить, что встроенный редактор поддерживает механизм создания стилей, использует набор шрифтов операционной системы, обеспечивает создание иерархии документов, связывая их гипертекстовыми ссылками, и многое другое. Созданные в этой среде курсовые проекты, дипломы или отчеты являются «живыми моделями», изменяющими свой вид при внесении в них корректив. наподобие того, как это происходит в электронных таблицах.

3. ИНТЕРФЕЙС MAPLE

Первое с чем встретится пользователь при загрузке Maple – это оконный интерфейс. Экранный интерфейс **Maple** состоит из следующих основных частей:

- строка команд;
- строка пиктограмм;
- рабочее окно, в котором производятся все математические операции действия, связанные с форматированием документа.

Строки команд будут несколько отличаться в зависимости от следующих действий:

А). Редактирование рабочего документа – стандартный интерфейс рабочего листа. Он будет показан на экране, если пользователь работает в рабочем документе и курсор ввода расположен именно там.

Б). Просмотр справки – интерфейс справочной системы. Гипертекстовые ссылки и скрытые секции делают общение с новой версией Browser лёгким, доступным и удобным.

Г). Двухмерные построения – интерфейс графической двумерной системы. При выполнении любых видов графических построений на плоскости перед пользователем появляется интерфейс двумерной графической системы. Если в пункте меню *Options* рабочего документа режим *Plot Display* установлен в значение *Windows*, то график будет строиться в отдельном окне. Если режим *Plot Display* установлен в значение *Inline*, то графические построения будут происходить непосредственно в рабочем документе.

Д). Трёхмерные построения - интерфейс графической трёхмерной системы, возникающий при любом виде трёхмерных изображений.

Нет необходимости приводить подробное описание всех пунктов меню перечисленных интерфейсов, так как это вполне понятно из названий самих пунктов меню.

4. СИНТАКСИС ЯЗЫКА MAPLE

Синтаксис Maple очень напоминает синтаксис таких языков программирования, как Паскаль и Фортран.

4.1 Символы и переменные

Любую команду можно заканчивать символом “:” или “;”. В первом случае команда будет выполнена, но результат не будет выведен на экран. Во втором случае ответ будет отображён на экране. При определении выражения используются стандартные символы: “+”, “-“, “*”, “/”, “:=”, “=”, “!”. Возвести выражения в степень можно двумя способами: “^” и “**”.

Для обозначения последовательности чисел используется символ “\$”

> x! \$x=1..4;

1, 2, 6, 24

Символ “@” – композиционный оператор. Например, чтобы вычислить вторую производную необходимо написать следующее:

> (D@@2)(ln);

$$a = -\frac{1}{a^2}$$

Численный параметр после символа “@” может принимать и отрицательные значения:

```
> (sin@@@(-1))(x);
arcsin(x)
```

Интересную роль играет символ «%» (процент). Одинарные процент ссылаются на результат предыдущей команды. Двойной - на результат, полученный две команды назад, и т.д. В качестве примера решим численным методом систему линейных уравнений, очистив предварительно память Maple и определив точность:

```
> restart: Digits:=2:
> 2*x+5*y-z=2:
> x+2*y=5:
> 2*x-z=4!:
> fsolve({%,%%,%}, {x,y,z});
{x=14.0, z=4.0, y=-4.5}
```

Здесь решение выводится в форме множества. Более подробно посмотреть в «Типы данных».

Существует также ряд команд для выполнения операций над множествами:

```
union    – объединение множеств;
intersect – пересечение множеств;
minus    – вычитание множеств;
```

Стандартные логические операции:

```
and – логическое "и"
or  – логическое "или"
not – логическое отрицание
```

Переменные в Maple характеризуются именем и типом. В качестве имени переменной может использоваться любой набор символов латинского алфавита, не зарезервированных программой. Нужно отметить, что система различает заглавное и строчное написание букв.

4.2 Константы и внутренние функции

Константы в Maple бывают целочисленными, числами с плавающей запятой и обыкновенными дробями. Кроме этих типов констант существуют символьные константы – зарезервированные имена. Например, false, true, Pi, I, и т.д.

Так можно (>False:=1;), а так -> (>>false:=1;) нельзя.

В Maple используются общепринятые среди математиков названия для основных математических функций, хотя и есть некоторые исключения.

ФУНКЦИЯ	ОПИСАНИЕ
abs	модуль
Re	действительная часть
Im	мнимая часть
factorial	факториал
log	обыкновенный логарифм
ln	натуральный логарифм
log10	десятичный логарифм
sqrt	квадратный корень
exp	экспонента
argument	аргумент комплексного числа
binomial	биномиальный коэффициент
round	округление
trune	отсечение дробной части

Тригонометрические функции записываются в форме, которая интуитивно понятна пользователю: $\sin$, $\cos$, $\tan$, $\sec$, $\csc$ и т.д.

В Maple запрограммированы некоторые математические функции, такие как гамма – функция, функция Лапласа, бета – функция, функция Бесселя, функция Дирака и Хэвисайда, функции Якоби и многие другие.

4.3 Типы данных

Около ста зарезервированных имён типов данных можно встретить на необъятных просторах Maple. Существует большое разнообразие функций для работы с данными, в структуре последних можно просто запутаться. Чтобы хоть как-то пролить свет на типы данных, рассмотрим основные, с которыми можно встретиться при выполнении различных вычислений.

4.3.1 Целые

В Maple выражение принадлежит к целому типу (integer), если оно состоит из последовательности цифр, не разделённых между собой никакими знаками. Длина последовательности может быть более 500000 цифр. С целыми числами возможны следующие операции:

```
>abs(-10420);      модуль числа;
10420
>factorial(5); 5!;  нахождение факториала;
120
120
```

Для проверки принадлежности к определённому типу служит команда `type`. Формат команды:

`Type(x, t)`, где x – любое выражение, t – название типа.

Например:

```
>type(-102, integer);
```

true

4.3.2 Дробные

Тип `fraction` - дробный тип. Дробь представляется в виде a/b , где a – целое число со знаком, b – целое число без знака. В выражении типа `fraction` обязательно присутствие двух полей: числитель и знаменатель. Функция `op` от дроби возвращает два числа – числитель и знаменатель.

```
>op(2/7);
```

2, 7

4.3.3 Числа с плавающей точкой

Тип `float` – числа с плавающей точкой. Тип `float` в среде Maple определён как:

1. последовательность чисел разделённых точкой:

а) `<integer>.<integer>`

б) `<integer>.`

в) `.<integer>`

2. число может быть представлено в виде:

`Float(mantissa, exponent)` т.е. `<mantissa>*10^<exponent>`

```
>type(.1234,float);
```

true

```
>Float(2,4);
```

20000.

Обратное представление числа реализуется функцией `op`, которая возвращает два числа – мантиссу и экспоненту.

```
>op(0. 02234);
```

2234, -5

Для приближения чисел с плавающей точкой служит команда `evalf`:

```
>evalf(Pi, 5);
```

3,1416

4.3.4 Строковые типы

В Maple определены тип `string` и тип `name`. Выражение типа `string` может содержать цифры и буквы, строчные и прописные. Строка с двух сторон должна быть окружена символом «'». Если же в строку требуется вставить символ «'», то его надо удвоить.

```
>var:= ' It' ' s a string ';
```

var:= It's a string

Из строки можно выделить подстроку:

```
>substring(abcdefgh, 3. .5);
```

cde

Определим длину строки:

```
>length(abcdef);
```

4.3.5 Булевы выражения

Для логических операций в среде Maple предусмотрен специальный тип данных – `boolean`, а зарезервированные слова `true` и `false` используются для работы с булевыми выражениями. В булевых выражениях используются следующие операторы: `'=','<','<=','end','or','not'`.

Для работы с булевыми выражениями предусмотрена команда `evalb` – вычисление сложного логического выражения:

```
>evalb(f=f);
true
>ToBe or not ToBe;
true
```

4.3.6 Последовательности

Последовательность – это набор элементов, разделённых запятыми, без скобок.

```
>S:=1,2,3,4,5,6,7;
S:=1,2,3,4,5,6,7
```

Для генерации последовательности в среде Maple служит команда `seq`.

```
seq(y, i=m..n)
```

```
seq(y, i=x),
```

где `y` – любое выражение; `i` – имя; `m,n` – численные параметры; `x` – выражение.

Наиболее распространённый вызов: `seq(f(i), i= 1..n)`, который генерирует последовательность `f(1),f(2),...,f(n)`.

Менее употребимый – `seq(f(i), i=m..n)` – создает последовательность `F(m), f(m+1), f(m+2),...,f(n)`. Здесь `m` и `n` могут быть не только типа `integer`.

```
>seq(sin(Pi*i/6), i=0..3);
```

$$0, \frac{1}{2}, \frac{1}{2}\sqrt{3}, 1$$

4.3.7 Множества

Множества принято обозначать фигурными скобками. Для них присущи все правила преобразования, принятые в классической математике.

```
>set1:={sin,cos,tan,cos};
```

```
set1:={sin,cos,tan}
```

Извлечение элементов с первого по второй из множества `set1`:

```
>op(1..2,set1);
```

```
cos, sin
```

Определение количества элементов в множестве `set1`:

```
>nops(set1);
```

```
3
```

Объединение двух множеств `{a, b}` и `{b, c}`:

```
>{a, b} union {b,c};
```

$\{a, c, b\}$

Пересечение:

```
>{a, b} intersect {b, c};
```

 $\{b\}$

Вычитание:

```
>{a, b} minus {b,c};
```

 $\{a\}$

Принадлежность элементов 'a' и 'cos' множеству set1:

```
>member(a, set1);
```

 $false$

```
>member(cos, set1);
```

 $true$

Зададим множество L в виде последовательности:

```
>L:= {seq{y[i], i=1..4}};
```

 $L := \{y_1, y_2, y_3, y_4\}$

Добавление элемента к множеству L:

```
>L:={op(L), z[5]};
```

 $L := \{y_1, y_2, y_3, y_4, z_5\}$

Удаление второго элемента из множества L:

```
>L:=subsop(2=NULL,L);
```

 $L := \{y_1, y_3, y_4, z_5\}$

4.3.8 Списки

Список принято обозначать квадратными скобками.

```
>list1:= [sin,cos,tan,cos];
```

 $list1 := [\sin, \cos, \tan, \cos]$

Со списками можно проводить математические операции, например, дифференцирование:

```
>D(list1);
```

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

По отношению к спискам и множествам допустимы операции присваивания:

```
>list2:=list1;
```

 $list2 := [\sin, \cos, \tan, \cos]$

4.3.9 Массивы

Массив – конечномерный список с целочисленными индексами.

array – создание массива;

print – распечатка содержимого массива;

map – задание операции над всеми элементами массива;

op – извлечение элементов(уточнение задания массива);

Создадим массивL

```
>v:=array(1..4);
```

```
v:=array(1..4, [ ])
```

Заполним этот массив элементами и распечатаем его содержимое:

```
>for i to 4 do v[i]:=i od;
```

```
>print(v);
```

```
[1,2,3,4]
```

Создадим одномерный массив 's' с нулевыми значениями:

```
>s:=array(1..2, [0,0]);
```

```
s:=[0, 0]
```

Создадим двумерный массив 'm':

```
>m:=array(symmetric, 1..2,1..2,[[cos(y),0],[0,sin(y)]]);
```

$$m := \begin{bmatrix} \cos(y) & 0 \\ 0 & \sin(y) \end{bmatrix}$$

Зададим операцию дифференцирования над всеми элементами массива 'm':

```
Mao(diff,m,y);
```

$$\begin{bmatrix} -\sin(y) & 0 \\ 0 & \cos(y) \end{bmatrix}$$

4.3.10 Таблицы

В отличие от массивов, где индексы целочисленные значения расположенные по порядку номеров, индексы у таблиц – любые значения.

Команда "table()" создает таблицу с неопределёнными значениями:

```
>table();
```

```
table([...])
```

Можно определить значения таблицы, выполнив команду:

```
>table([22,42]);
```

```
table([1 = 22, 2 = 42])
```

Здесь программа сама присвоила целочисленные значения, расположенные по порядку.

Индексы таблицы можно задать произвольным образом:

```
>S:= table([(red)=45, (green)=61]);
```

```
S:=table([green = 61, red = 45])
```

Можно обратиться и к элементам таблицы:

```
S[1], S[red];
```

```
S1, 45
```

Зададим таблицу F, элементами которой являются операторы:

```
>F:=table([y=(x->x^2),cos=-sin]);
```

Распечатаем таблицу:

```
print(F);
```

```
table([y=x->x^2, cos=-sin])
```

Вычислим значения элемента таблицы 'y' от аргумента, равного 3:

```
>F[y](3);
```

5. ОСНОВНЫЕ МАТЕМАТИЧЕСКИЕ ОПЕРАЦИИ

5.1 Операции с формулами

Maple позволяет сосредоточиться на основной нити проводимых преобразований, благодаря имеющейся возможности избегать рутинной работы, такой как, раскрытие скобок, приведение подобных, разложение на множители, и т.д. Приведём основные команды этого класса.

КОМАНДА	ОПИСАНИЕ
collect(w,x)	Приведение подобных членов в выражении w относительно переменной x
denom(d)	Выделение знаменателя дроби d
expand(w)	Раскрытие скобок выражения w
factor(w)	Факторизация (разложение на множители) выражения w
lhs(ur)	Выделение левой части уравнения ur
normal(w)	Нормализация (сокращение) дроби w
numer(w)	Выделение числителя дроби d
op(i..j,e)	Выделение подвыражения из выражения e
rhs(ur)	Выделение правой части уравнения ur
simplify(w)	Упрощение выражения w
subs(x=t, w)	Подстановка в выражение w вместо выражения x выражения t
subsop(eg1,...,egN,expr)	Замена некоторого операнда в выражении expr
trigsb(w)	Определение всех тригонометрических эквивалентов выражения w

Рассмотрим операцию упрощения выражения. Присвоим переменной **rex** некоторую сумму из тригонометрических слагаемых.

```
>rex:=cos(x)^3+sin(x)^4+2*cos(x)^4-2*sin(x)^4-cos(2*x);
```

Упростим полученный полином:

```
>simplify(rex);
```

$$\cos(x)^3 + \cos(x)^4$$

Для раскрытия скобок используем команду *expand*, а затем определим дробь.

```
>expand((x+3)*(x-y)): expand((x+3)*(x+1));
```

```
>w2:=(%)/(%%);
```

$$w2 := \frac{x^2 + 4x + 3}{x^2 - xy + 3x - 3y}$$

А теперь произведём факторизацию выражения **w2** (операцию обратную *expand*).

```
>factor(w2);
```

$$(x+1)/(x-y)$$

Для выделения подвыражения из целого выражения служит команда *op*.
Формат команды:

op(*i*,*e*), *op*(*i*..*j*, *e*), *op*(*e*), где *i*, *j* – положительные целые числа, определяющие позицию операнда в выражении, *e* – любое выражение.

>*f*:=*[x, y, z]*;

$$f := [x, y, z]$$

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

z

Для замены некоторого операнда в выражении служит команда *subsop*.
Формат команды:

subsop(*eq1, eq2,...,eqN, expr*), где *eqI* – выражение (необязательное) вида: <*numI*>=<*exprI*>, *numI* – положительное целое, *exprI* – выражение, *expr* – выражение.

Заменяем в определённом нами ранее списке *f*=[*x, y, z*] третий элемент этого списка на *v*:

>*subsop*(3=*v*, *f*);

$$[x, y, z]$$

Если необходимо подставить одно выражение в другое, а также проверить полученное решение путем подстановки его в исходное равенство нужно воспользоваться командой *subs*. Формат команды:

subs(*s_1, s_2,..., s_n, expr*), где *s_1,...* - уравнение, или множество, или список из уравнений, *expr* – любое выражение. При этом *s_1,...,s_n* подставляются в выражение *expr*. Например:

>*subs*(*x= r ^*(1/3), 3**x**ln(*x*^3));

$$3r^{\frac{1}{3}} \ln(r)$$

Другой пример использования функции *subs* – проверка полученного решения.

>*egs*:=*{2*x*y=1, x+z=0, 2*x-3*z=2}*;

$$egs := \{2xy=1, x+z=0, 2x-3z=2\}$$

>*f*:=*solve*(*egs, {x, y, z}*);

$$f := \left\{ x = \frac{2}{5}, z = \frac{-2}{5}, y = \frac{5}{4} \right\}$$

Проверим правильность решения системы:

>*subs*(*f, egs*);

$$\{1=1, 2=2, 0=0\}$$

Для выделения правой и левой частей выражения очень удобны операторы `rhs` и `lhs`:

```
>y=a*x^2+b;
```

$$y = ax^2 + b$$

```
>rhs(%);
```

$$ax^2 + b$$

```
>lhs(%%);
```

$$y$$

Часто полезно бывает представить некоторую функцию всеми её тригонометрическими эквивалентами. Для этой цели служит функция *trigsubs*. Перед использованием её необходимо подгрузить командой *readlib*.

```
>readlib(trigsubs):
```

```
>trigsubs(sin(2*a);
```

$$\left[\sin(2a), 2\sin(a)\cos(a), \frac{1}{\csc(2a)}, 2\frac{\tan(a)}{1+\tan(a)^2}, -\frac{1}{2}I(e^{(2Ia)} - e^{(2Ia)}) \right]$$

5.2 Операции с полиномами

Под полиномом Maple понимает сумму или разность выражений с неотрицательными степенями. В таблице приведены основные команды для работы с полиномами.

КОМАНДА	ОПИСАНИЕ
lcoeff(pol, opt)	Определение старшего коэффициента полинома pol
coeff(pol, x, n)	Определение коэффициента полинома pol при n – ой степени переменной x
coeffs(pol, x)	Определение всех коэффициентов полинома pol при переменной x
convert(pol, sqrfree, x)	Разложение полинома pol на квадратные трёхчлены по переменной x
degree(pol, x)	Определение степени полинома pol при переменной x
discrim(pol, x)	Вычисление дискриминанта полинома pol по переменной x
gcd(pol1, pol2)	Вычисление наибольшего общего делителя двух полиномов pol1 и pol2
psqrt(pol)	Вычисление квадратного корня из полинома pol
quo(pol1, pol2, x)	Вычисление частного от деления двух полиномов pol1 и pol2 по переменной x
randpoly(x)	Создание случайного полинома переменной x
rem(pol1, pol2, x)	Вычисление остатка от деления двух полиномов pol1 и pol2 по переменной x

Определим полином **pol**:

```
>pol:=expand((5*y*x^2+x+1)*(x^3-x)+(2*y*x^2+6);
```

$$pol := 5yx^5 - 5yx^3 + x^4 - x^2 + x^3 - x + 2yx^2 + 6$$

С помощью команды **quo** разделим один полином на другой, определив при этом целую часть от деления:

```
>quo(pol, x^3-x, x);
```

$$5yx^2 + x + 1$$

С помощью команды **rem** найдём остаток от этого деления,. При этом целая часть запишется в переменную **`k`**:

```
>os:=rem(pol, x^3-x, x, `k`); k;
```

$$os := 2yx^2 + 6$$

$$5yx^2 + x + 1$$

Для нахождения наибольшего общего делителя двух полиномов используем команду **gcd**.

```
>gcd(os, y*x^2+3);
```

$$yx^2 + 3$$

Сгруппируем ранее определённый нами полином по переменной x с помощью команды *collect* и подставим вместо переменной y число 2.

```
>collect(pol, x); pol1:=subs(y=2, pol);
```

$$5yx^5 + x^4 + (1-5y)x^3 + (-1+2y)x^2 - x + 6$$

$$pol1 := 10x^5 - 9x^3 + x^4 + 3x^2 - x + 6$$

Определим коэффициенты полинома $pol1$ и далее вычислим для него дискриминант и степень, используя соответственно команды *coeffs*, *discrim*, *degree*.

```
>coeffs(pol1, x); discrim(pol1, x); degree(pol1, x);
```

```
10, -1, 1, -9, 3, 6
```

```
63388974740
```

```
5
```

5.3 Ограничение на переменные

В Maple можно наложить различные ограничения на переменные, при этом программа будет предупреждать вас о наложенных ограничениях в различной форме. Пусть переменная “ a ” будет больше нуля:

```
>assume(a>0); signum(a);
```

```
1
```

```
>re(a+1);
```

```
a~+1
```

После переменной “ a ” мы видим символ “ $\sim$ ”, означающий, что на переменную наложены ограничения. Предупреждающая надпись может быть и такого рода (если установить пункт меню Options/Assumed Variables/Phrase):

```
>assume(n, integer); cos(n*Pi);
```

$$(-1)^n$$

with assumptions on n

6. НЕКОТОРЫЕ ПРИМЕРЫ МАТЕМАТИЧЕСКОГО АНАЛИЗА

Maple позволяет сэкономить время и избежать многих ошибок при вычислении таких понятий математического анализа как пределы, производные, первообразные функций, интегралы, ряды, дифференциальные уравнения.

Рассмотрим на примерах команду *limit*, которая позволяет находить пределы функций.

```
>restart; f(x):=(x^3-3*x^2+2*x-5)/(x^2+2);
```

```
>Limit(f(x), x=-1)=limit(f(x), x=-1);
```

$$\lim_{x \rightarrow (-1)} \frac{x^3 - 3x^2 + 2x - 5}{x^2 + 2} = -\frac{11}{3}$$

```
>r:=5*sin((3*x)/(x-Pi)); limit(r, X=Pi);
```

-5..5

Последний результат означает, что предел не найден, но значения функции в окрестности указанной точки принимают значения в диапазоне, который вычислила функция `limit`. Проверить это можно с помощью графика функции. Например:

```
>plot(r, x=Pi-0.3..Pi+0.3);
```

Иногда Maple не может найти предел, например:

```
>limit(tan(x), x=infinity);
```

undefined

Можно находить односторонние пределы. Для этого достаточно указать ключевое слово `left` или `right`.

```
>g(x):=1/x: Limit(g(x), x=0, left)=limit(g(x),x=0, left);
```

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

проверить правильность нахождения предела можно с помощью графика.

```
>plot(g(x), x= -1..1, view=[-1..1, -50..50],discont=true, color=blue, labels=['x`, `g(x)`]);
```

Можно также находить пределы для функций нескольких аргументов:

```
Limit(x+1/y, {x=0, y=infinity});
```

0

Для Maple не составит труда найти предел для функции с неизвестными параметрами:

```
>limit(a*x, x=infinity);
```

signum(a) ∞

Чтобы продифференцировать функцию, достаточно воспользоваться командой `diff`:

```
>restart: f(x):=ln(sqrt(exp(3*x)/(1+exp(3*x)))): simplify(diff(f(x), x));
```

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

Можно взять частные производные от функции многих переменных:

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

$$\frac{\partial^2}{\partial y \partial x} f(x, y)$$

При помощи оператора формирования последовательности (\$) можно брать производные высоких порядков.

```
>Diff(sin(x), x$3)=diff(sin(x), x$3);
```

$$\frac{\partial^3}{\partial x^3} \sin(x) = -\cos(x)$$

Взять интеграл от какой-либо функции можно при помощи оператора `int`.

```
>restart:
```

Неопределённый интеграл:

```
>Int((3*x^2+8)/(x^3+4*x^2+4*x), x)=int((3*x^2+8)/(x^3+4*x^2+4*x), x);
```

$$\int \frac{3x^2 + 8}{x^3 + 4x^2 + 4x} dx = 2 \ln(x) + \frac{10}{x+2} + \ln(x+2)$$

Определённый интеграл:

>Int(sin(phi)^3*sqrt(cos(phi)), phi=0..Pi/2)=int(sin(phi)^3*sqrt(cos(phi)), phi=0..Pi/2);

$$\int_0^{\frac{1}{2}\pi} \sin(\phi)^3 \sqrt{\cos(\phi)} d\phi = \frac{8}{21}$$

Несобственные интегралы (первого и второго рода):

>Int(1/(x^2+2*x+2), x=-infinity..infinity)=int(1/(x^2+2*x+2), x=-infinity..infinity);

$$\int_{-\infty}^{\infty} \frac{1}{(x^2 + 2x + 2)} dx = \pi$$

>Int(1/(x-1)^2, x=0..2)=int(1/(x-1)^2, x=0..2);

$$\int_0^2 \frac{1}{(x-1)^2} dx = \infty$$

В тех случаях, когда интеграл не может быть вычислен в численной форме, поможет функция evalf.

>ww:=int(exp(-x^3), x=0..1);

$$ww := \int_0^1 e^{(-x^3)} dx$$

>evalf(ww, 5);

.80751

В Maple можно при помощи команды sum находить предел сходимости ряда.

>sum(7^(3*n)/(2*n-5)!, n=3..infinity)=evalf(sum((7^(3*n)/(2*n-5)!, n=3..infinity));

$$\sum_{n=3}^{\infty} \frac{7^{(3n)}}{(2n-5)!} = .1203515449 * 10^{15}$$

Полезной может оказаться и функция product:

>product(a[k], k=0..n);

$$\prod_{k=0}^n a_k$$

Достаточно просто производить различные операции над кусочно-аналитическими функциями. Их можно интегрировать, дифференцировать, и даже использовать их в дифференциальных уравнениях.

>p:=piecewise(x<0, -1, x>1, 2*x, x^2);

$$\begin{cases} -1 & x < 0 \\ 2x & 1 < x \\ x^2 & \text{otherwise} \end{cases}$$

Построим график вышеописанной функции:

```
>plot(p, x= -1..2);
```

Интегрировать и дифференцировать кусочную функцию можно также, как и обычные функции:

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

$$\begin{cases} -x & x \leq 0 \\ \frac{1}{3}x^3 & x \leq 1 \\ x^2 - \frac{2}{3} & 1 \leq x \end{cases}$$

А теперь попробуем решить дифференциальное уравнение, в которое входит функция $p(x)$.

```
>dsolve({diff(y(x), x)+p*y(x), y(0)= -2}, y(x));
```

$$y(x) = \begin{cases} -2e^x & x \leq 0 \\ -2e^{(\frac{1}{3}x^3)} & x \leq 1 \\ -2e^{(-x^2 + \frac{2}{3})} & 1 < x \end{cases}$$

7. ИНТЕГРАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ

Интегральные преобразования особенно полезны при решении дифференциальных уравнений. Библиотека `inttrans` содержит наиболее часто встречающиеся функции. Выражение над которым происходит преобразование, может включать в себя полиномы, экспоненты, тригонометрические функции, дельта-функцию Дирака ($\text{Dirac}(t)$), функцию Хевисайда ($\text{Heaviside}(t)$), а также интегралы и производные. Кроме того, предусмотрена возможность расширения таблицы интегралов равенствами пользователя. Это достигается с помощью функции `addtable`.

После подключения библиотеки `inttrans` командой `with(inttrans)` становятся доступными следующие функции:

КОМАНДА	ОПИСАНИЕ
<code>addtable(tname, f(t), F(s), t, s)</code>	Добавление к таблице интегральных преобразований (<code>tname=fourier, invlaplace,...</code>) собственных равенств
<code>fourier(f(t), t, u)</code>	Преобразование Фурье функции f(t) относительно переменной t
<code>fouriercos(f(t), t, u)</code>	Тригонометрическое преобразование Фурье по косинусу функции f(t) относительно переменной t
<code>fouriersin(f(t), t, u)</code>	Тригонометрическое преобразование Фурье по синусу функции f(t) относительно переменной t
<code>hankel(f(t), t, h)</code>	Преобразование Ханкеля функции f(t) относительно переменной t
<code>Hilbert(f(t), t, h)</code>	Преобразование Гильберта функции f(t) относительно переменной t
<code>invfourier(f(u), u, t)</code>	Обратное преобразование Фурье функции f(u) относительно переменной u
<code>invhilbert(f(h), h, t)</code>	Обратное преобразование Гильберта функции f(h) относительно переменной h
<code>invlaplace(f(s), s, t)</code>	Обратное преобразование Лапласа функции f(s) относительно переменной s
<code>laplace(f(t), t, s)</code>	Преобразование Лапласа функции f(t) относительно переменной t
<code>mellin(f(t), t, m)</code>	Преобразование Меллина функции f(t) относительно переменной t

Для выполнения стандартного преобразования Лапласа над функцией $f(t)$ относительно переменной t используется команда $\text{laplace}(f(t), t, s)$, где s – параметр.

$>\text{laplace}(t^3 + \cos(t) = y(t), t, s);$

$$\frac{6}{s^4} + \frac{s}{s^2 + 1} = \text{laplace}(y(t), t, s)$$

Выполнив обратное преобразование Лапласа, получаем исходный результат.

$\text{invlaplace}(\%, s, t);$

$$t^3 + \cos(t) = y(t)$$

Тригонометрические преобразования Фурье выполняются следующей командой:

$>\text{fouriercos}(1/t^2 + 3), t, s);$

$$\frac{1}{6} \sqrt{2} \sqrt{\pi} \sqrt{3} e^{(-\sqrt{3} s)}$$

$>\text{fouriersin}(f(t), t, s);$

$$\text{fouriersin}(f(t), t, s)$$

Из последнего выражения мы видим, что Maple не смог преобразовать функцию $f(t)$, т.к. не нашёл соответствующего правила в встроенной таблице интегралов. Мы можем определить сами изображение данной функции и записать его в таблицу Maple.

$>\text{addtable}(\text{fouriersin}, f(t), F(s), t, s);$

Теперь снова выполнив, предыдущее преобразование мы видим, что определённое нами правило включено в таблицу и сним можно выполнять различные операции.

$>\text{fouriersin}(f(x), x, z);$

$$F(z)$$

8 ГРАФИКА MAPLE

В Maple можно размещать графику сразу в рабочем документе. Разумеется при необходимости её можно построить и в рабочем окне. После построения графика параметры изображения можно изменить с помощью специального меню пиктограмм или всплывающего меню, которое вызывается нажатием правой кнопки мыши.

С помощью «мыши» можно узнать точные координаты интересующей точки, просто указав на неё и нажав левую кнопку. А при построении трёхмерных графиков можно визуальнo «крутить» оси координат, просматривая сложные участки поверхности с разных точек зрения.

8.1 Графика 2D.

8.1.1 Задание областей

Область – это окно декартовой системы координат, в котором строится график.

Синтаксис определения области:

x = нижняя граница..верхняя граница.

Числа, определяющие границы, должны быть действительными.

Plot(f, x = low..hi, y = low..hi) – пример задания.

Области можно задавать с использованием констант.

Например: *infinity*, *Pi*, *exp(8)*, и т.д. По умолчанию выбирается диапазон-10..10 для оси абсцисс. Если указан один диапазон, то считается, что он для оси абсцисс, а для оси ординат область изменения выбирается автоматически.

```
>plot(sin(x), x=0..Pi, y=0..0.5);
```

Или

```
>plot(exp, 0..infinity);
```

8.1.2 Стили

При построении можно выбрать стиль (тип) интерполирования. Задаётся стиль с помощью ключевого слова *style*: *plot(f, h, v, style=x)*.

Существуют три стиля:

POINT - построение по точкам;

LINE - линейная интерполяция;

PATCH - стиль для многоугольников.

Стиль *point* – график будет строиться по точкам. Точки могут быть заданы парами в виде списка: $[[x_1, y_1], [x_2, y_2], \dots, [x_n, y_n]]$

```
>plot(x^3, x= - 1..1, style=point);
```

Стиль *line* – точки будут соединяться прямыми. Данный стиль выбирается по умолчанию.

Стиль *patch* – применяется для построения раскрашенных многоугольников.

```
>plot([seq([cos(2*Pi*i/5), sin(2*Pi*i/5)], i=1..5), [cos(2*Pi/5), sin(2*Pi/5)]],  
style=patch, color=green);
```

8.1.3 Параметры

Параметры перечисляются в команде plot после указания областей в форме

<имя параметра>=<значение>.

Список параметров приведен в таблице.

ПАРАМЕТРЫ	ЗНАЧЕНИЯ	ОПИСАНИЕ
adaptive	true, false	использование адаптивного алгоритма построения
axes	FRAME, BOXED, NORMAL, NONE	тип координатных осей
axesfont	[family, style, size]	шрифт для осей
color	Зарезервированное слово или процедура	цвет графика
cords	имя системы координат	тип системы координат
discont	true, false	для построения выражений с разрывами
font	[family, style, size]	шрифт для текста
labelfont	[family, style, size]	шрифт для меток осей
labels	[str1, str2]	названия осей
linestyle	целое число	тип линий
numpoints	целое число	точки по оси абсцисс
resolution	целое число	горизонтальное разрешение устройства вывода
sample	[x1, x2,...,xn]	Список точек в которых будет построена функция (adaptive=false)
scaling	CONSTRAINED, UNCONSTRAINED	масштабирование
style	POINT, LINE, PATCH	тип интерполяции
symbol	BOX, CROSS, CIRCLE, POINT, DIAMOND	символ точек чертежа
thickness	0, 1, 2, 3	толщина линий
title	строка	заголовок чертежа
titlefont	[family, style, size]	шрифт для заголовка
view	[x1..x2, y1..y2]	окно координатной плоскости
xtickmarks ytickmarks	целое число	количество отметок на осях X и Y

Пример:

>plot(sin, sample=[0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9], adaptive=false);

8.1.4 Кусочные функции

Для построения кусочной функции надо просто определить описывающую её процедуру, а затем как обычно воспользоваться командой `plot`.

```
>w:=proc(x) if x<0 then - x elif (x>0) and (x<4) then x else - x+8 fi end;
>plot(w, -5..6, color=red);
```

8.1.5 Параметрическая графика

При построении параметрических функций используется следующий синтаксис команды `plot`:

```
Plot([x(t), y(t), t=(диапазон изменения t], h, v, options)
>plot([(t^2-1)/(t^2+1), 2*t/(t^2+1), t= - infinity..infinity]);
```

8.1.6 Построение графиков в различных системах координат

При необходимости можно выбрать систему координат, отличную от декартовой. Всего Maple может использовать 15 типов систем координат (для двумерного построения), которые задаются параметром `coords="имя координат"`.

Приведем пример построения в полярных координатах:

```
>plot([1-cos(t), t, t=0..2*Pi], coords=polar);
```

8.1.7 Анимация 2D графиков

В Maple возможна анимация двумерных графиков. Причем, в отличие от предыдущих версий это можно сделать не только в отдельном окне, но и в самом документе. При этом пользователь может продолжать свою работу!

```
>with(plots):
```

```
>animate({x-x^3/u, sin(u*x)}, x=0..Pi/2, u=1..16, color=red);
```

Синтаксис команды `animate`:

```
Animate({F, x, t,...}).
```

Здесь $F=F(x, t)$ – функция двух переменных; x, t – диапазоны изменения x и t .

При анимации происходит следующее: изменяются значения t и при фиксированных значениях t строится график $F(x, t)$. Количество выводимых кадров можно устанавливать параметром `frames` (по умолчанию `frames = 16`).

8.1.8 Совмещение графиков

Можно построить несколько графиков на одной координатной плоскости. Для этого достаточно указать в команде `plot` множество или список функций, т.е. записать через запятую функции и заключить их в фигурные или квадратные скобки. В этом случае Maple автоматически выбирает разные цвета для графиков.

Допустимо совмещать обычную и параметрическую графику.

```
>plot({x, [x^2, x, x=0..1]}, x=0..1);
```

При необходимости для каждой функции можно указать конкретный цвет и стиль построения.

```
>plot([cos(x), cos(x+0.1)], x=0..2, color=[red, blue], style=[point, line],  
symbol=diamond);
```

Уважаемый читатель, данное методическое пособие является лишь вводным курсом в Maple. Для того, чтобы продолжить дальнейшее совершенствование вам необходимо будет обратиться дополнительно к специализированной литературе.

В заключение приведём обзор библиотек Maple.

9. ОБЗОР БИБЛИОТЕК MAPLE

DEtools	Средства работы с дифференциальными выражениями
Domains	Библиотека для создания сложных алгоритмов
GF	Поля Галоиса
GaussInt	Гауссовы целые
LRtools	Работа с линейными рекуррентными выражениями
combinat	Комбинаторика(функции)
combstruct	Комбинаторика (структуры)
diffforms	Дифференциальные формы
finance	Финансовая математика
geometry	Евклидова геометрия
grobner	Базис Гробнера
group	Теория групп
intrans	Интегральные преобразования
liesymm	Симметрия Ли
linalg	Линейная алгебра
logic	Булева логика
networks	Теория графов
numapprox	Численная аппроксимация
numtheory	Теория чисел
orthopoly	Ортогональные полиномы
padic	p – адические числа
plots	Графическая библиотека
plottools	Графические примитивы
powseries	Степенные ряды
process	Мультизадачность (Unix)
simplex	Оптимизация линейных систем
stats	Статистика
student	Общеупотребительные математические функции
sumtools	Суммы и ряды
tensor	Тензоры
totorder	Наложение ограничений на переменные

10. ХОД РАБОТЫ

1. Запустить MAPLE.
2. Ознакомиться с назначением окон, панелей и кнопок Maple.
3. Начертить (*не копируя*) командное окно Maple и меню команд **File** с переводом на русский язык
4. Выполнить по одному примеру из каждого пункта настоящей методички.
5. Выполнить описание одной из указанных преподавателем библиотек Maple (назначение, возможности, ограничения).
6. Оформить отчёт. Защитить работу.

11. СОДЕРЖАНИЕ ОТЧЁТА

1. Цель работы.
2. Задачи Maple.
3. Результаты работы по пунктам 3, 4, 5 настоящих методических указаний.
4. Развернутый вывод по работе.

12. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое MAPLE?
2. Пояснить назначение элементов интерфейса Maple.
3. Синтаксис языка Maple.
4. Символы и переменные.
5. Константы и внутренние функции.
6. Типы данных.
7. Основные математические операции.
8. Интегральные преобразования.
9. Графика в Maple.
10. Обзор библиотек Maple.
11. Каким образом можно получить справку о той или иной команде.

СПИСОК ЛИТЕРАТУРЫ

1. Прохоров Г.В. «Математический пакет Maple V Release 4», руководство пользователя/ Прохоров Г.В., Колбеев В.В., Желнов К.И., Леденёв М.А. – СПб.: Калуга, «Облиздат», 1988г. – 272 с.

