Кафедра: Информационные Технологии

**Лабораторная Работа**

**На тему: ДИФФЕРЕНЦИРОВАНИЕ, ИНТЕГРИРОВАНИЕ, ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ, СУММ, РЯДОВ ФУНКЦИЙ И МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ.**

Москва, 2008 год

**ДИФФЕРЕНЦИРОВАНИЕ, ИНТЕГРИРОВАНИЕ, ВЫЧИСЛЕНИЕ ПРЕДЕЛОВ, СУММ, РЯДОВ ФУНКЦИЙ И МАТЕМАТИЧЕСКИХ ВЫРАЖЕНИЙ**

Цели работы:

• знать команды, используемые при вычислении обыкновенных и частных производных аналитического выражения по одной или нескольким переменным в системе вычислений Maple;

• знать команды, используемые при интегрировании аналитических выражений в системе вычислений Maple;

• знать команды, используемые при вычислении пределов, сумм, рядов функций в системе вычислений Maple;

• уметь применять указанные команды для решения математических задач.

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

1. **Дифференцирование выражений**

Команды diff ( ) и Diff ( ) предназначены для вычисления обыкновенных и частных производных аналитического выражения по одной или нескольким переменным. Вторая команда является отложенной командой, которая не вычисляет производную от выражения, а просто отображает математическую запись взятия производной. Результат действия отложенной команды можно присвоить переменной Maple, а в дальнейшем при помощи команды value ( ) вычислить результат этой отложенной команды. Отложенная форма команды удобна, когда необходимо видеть, какие операции были сделаны для получе­ния нужного выражения. Кроме этой команды еще целый ряд команд имеют отложенную форму, информацию о которых можно получить в Справке.

Синтаксис команды дифференцирования следующий:

diff (выражение, переменная\_1, переменная\_2, ..., переменная\_n);

diff (выражение, [переменная\_1, переменная\_2, ..., переменная\_n]);

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

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

Пример 1. Вычисление производных.

> s:=x^3\*cos(x)+y^2\*ln(sin(x));



> diff(s,x);



> diff(s,x$2);



> diff(s,x,y);



> fs:=Diff(s,x);



> q:=sqrt(fs);



> value(%);



Последние три команды показывают использование отложенной формы команды дифференцирования.

**2. Интегрирование выражений**

Команда int( ) имеет отложенную форму Int( ) и осуществляет интегрирование выражений по заданной переменной. Эта команда вычисляет неопределенный интеграл от выражения (при этом, правда, в ответе не будет никакой постоянной интегрирования) используя следующий синтаксис:

int (выражение, переменная);

Определенный интеграл вычисляется при следующем синтаксисе команды:

int (выражение, переменная = a..b);

где a и b являются пределами интегрирования, причем эти пределы могут быть и аналитическими выражениями.

Пример 2. Интегрирование функций.

> f:=x^2\*cos(k\*x);



> int(f,x);



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



> Int(f,x=0..Pi);



> value(%);



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

По умолчанию команда интегрирования проверяет выражение на не­прерывность в области интегрирования и вычисляет интеграл как сумму от­дельных определенных интегралов на промежутках непрерывности функции. Опция `continuous` отключает этот режим и вычисляет интеграл как разность значений первообразной подынтегральной функции в точке начала и конца промежутка интегрирования. Еще одна опция `CauchyPrincipalValue` вычисляет несобственные интегралы первого и второго рода в смысле главного значения Коши.

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

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

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

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

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

При численном интегрировании по умолчанию используется квадра­турная формула Кленшо-Куртиса (Clenshaw-Curtis). Если в подынтегральном выражении встречается сингулярность, то применяется специальная методи­ка символьного анализа для ее разрешения. Для задач с неустранимыми сингулярностями используется адаптивный метод двойных экспоненциаль­ных квадратур. Параметр flag позволяет явно задать метод численного интегрирования. Он может принимать значения, представленные в табл. 1.

Таблица 1. Значения параметра flag при численном интегрировании.

|  |  |
| --- | --- |
| Значение | Смысл |
| \_Ccquad | Применяется только квадратура Кленшо-Куртиса без вызова процедуры обработки сингулярности |
| \_Dexp | Применяется адаптивный метод двойных экспоненциальных квадратур |
| \_Ncrule | Применяется метод квадратурной формулы Ньютона-Котеса, являющийся методом фиксированного порядка, и не эффективен для высоких точностей (Digits > 15) |

Пример 3 помогает освоиться с использованием вышеприведенной методики.

Пример 3. Численное интегрирование функций.

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



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



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



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



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



Первый интеграл примера 3 вычисляется в аналитическом виде, но представляется через значение специальной функции интегральный косинус. Для получения ответа в виде десятичного числа применяется алгоритм чис­ленного интегрирования. Здесь же показано использование отложенной формы команды интегрирования для более удобного представления ответа.

Численное интегрирование даже функций, внешний вид которых представляется не достаточно сложным, может потребовать значительного времени. Если будет казаться, что Maple завис (а такое случается), то надо следить за изменением времени в правой части строки состояния. Если оно изменяется, то просто следует дождаться завершения интегрирования.

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

**3. Пределы.**

Для нахождения предела выражения или функции в Maple используется команда limit (параметр 1, параметр 2). Первый параметр – выражение, второй параметр — имя переменной, приравненное значению переменной в точке предела. Необязательный третий параметр – направление предела. Если направление не задано, вычисляется стандартный двусторонний предел. Если предел не существует, в качестве ответа возвращается сообщение "undefined". Если Maple не способен вычислить предел (однако он может существовать), возвращается невыполненная команда.

> limit(cos(x)/x,x=Pi/2);



> limit((-x^2+x+1)/(x+4),x=infinity);



> limit(tan(x),x=Pi/2);



В большом количестве случаев выражение, которое не имеет двустороннего предела, имеет односторонний предел:

limit(tan(x),x=Pi/2,left);



limit(tan(x),x=Pi/2,right);



limit((1+a/x)^x,x=infinity);



В команде limit ( ) может присутствовать также необязательная опция complex или real в качестве третьего параметра аргумента. Эта опция определяет, в комплексной или действительной области вычисляется предел.

Задачи для самостоятельного решения.

1. Найти производную:

1.1. ; 1.7. ;



1.2. ; 1.8. ;



1.3. ; 1.9.;



1.4. ; 1.10. ;



1.5. ; 1.11. ;



1.6. ; 1.12. ;



1.13. .



2. Найти интеграл:

2.1. , 2.7. ,



2.2. , 2.8. при ,



2.3. , 2.9. ,



2.4. , 2.10. при ,



2.5. , 2.11. при



при ,



2.6. , 2.12. ,



2.13. .



3. Найти следующие пределы:

3.1. ;



3.2. ;



3.3. ;



3.4. ;



3.5. ;



3.6. ;



3.7. ;



3.8. ;



3.9. ;



3.10. ;



3.11. ;



3.12. ;



3.13. .



Варианты заданий.

. 1.1; 2.1; 3.1; . 1.9; 2.9; 3.9;



. 1.2; 2.2; 3.2; . 1.10; 2.10; 3.10;



. 1.3; 2.3; 3.3; . 1.11; 2.11; 3.11;



. 1.4; 2.4; 3.4; . 1.12; 2.12; 3.12;



. 1.5; 2.5; 3.5; . 1.13; 2.13; 3.13;



. 1.6; 2.6; 3.6; . 1.1; 2.2; 3.3;



. 1.7; 2.7; 3.7; . 1.13; 2.12; 3.11.



. 1.8; 2.8; 3.8;



Контрольные вопросы.

1. Команда diff ( ), ее предназначение и синтаксис.

2. Команда int ( ), ее предназначение и синтаксис.

3. Отложенные формы команд diff ( ) и int ( ).

4. Формирование производных высокого порядка с помощью оператора последовательности $.

5. Как вычислить значение определенного интеграла численным способом?

6. Назначение параметра digits в команде evalf ( ).

7. Назначение параметра flag в команде evalf ( ).

8. Команда limit( ), ее предназначение и синтаксис.

**Литература**

1. Говорухин В.Н., Цибулин В.Г. Введение в Maple. Математический пакет для всех. – М.: Мир, 1997. – 208 с.

2. Дьяконов В.П. Математическая система Maple V. – М.: Издательство “Солон”,1998.

3. Двайт Г.Б. Таблицы интегралов и другие математические формулы. – М.: Наука. Главная редакция физико-математической литературы, 1983. – 176 с.

4. Матросов А.В. Maple 6. Решение задач высшей математики и механики. – СПб.:БХВ - Петербург, 2001.– 528 с.

5. Манзон Б.М. Maple V Power Edition – М.: Информационно-издательский дом “Филинъ”,1998г.