

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ»

В.В. Чернушкин, В.Д. Овсянников

**МОДЕЛИРОВАНИЕ ЗАДАЧ  
КВАНТОВОЙ МЕХАНИКИ  
В СРЕДЕ MAXIMA**

*Учебное пособие*

Воронеж  
Издательский дом ВГУ  
2016

## Содержание

Предисловие	4
Введение в среду Maxima	4
Встроенные константы. Переменные и функции	7
Аналитические преобразования алгебраических выражений	12
Суммы, произведения, пределы	16
Решение уравнений и систем уравнений	18
Операции с векторами и матрицами	20
Дифференцирование и интегрирование выражений	25
Решение дифференциальных уравнений	26
Построение графиков функций	28
Специальные функции	34
Расчет коэффициентов квантовой теории угловых моментов	40
Основы программирования в среде MAXIMA	44
Нахождение решения стационарного уравнения Шредингера для различных квантовых систем	51
Стоячие волны	52
Полупроницаемая перегородка	54
Прямоугольная потенциальная яма	59
Виртуальные уровни	61
Дираковская потенциальная гребенка (периодический потенциал)	65
Гармонический осциллятор	68
Частица внутри потенциальной сферы	70
Сферически симметричная прямоугольная яма конечной глубины	72
Задачи для самостоятельного решения	75
Библиографический список	76

историю команд (ввод и вывод), а файлы типа wxm сохраняют только команды ввода. После повторного считывания файлов нужно запустить на исполнение при помощи команд основного меню, например, Ячейка -> Evaluate All Cells (Ctrl+Shift+R).

Все команды вводятся в поле ввода, разделителем команд является символ ; (точка с запятой). Для создания окна ввода нужно нажать клавиши Shift+Enter. После ввода команды необходимо нажать клавиши Ctrl+Enter для её обработки и вывода результата. В ранних версиях Maxima и некоторых её оболочках (например, xMaxima) наличие точки с запятой после каждой команды строго обязательно. Завершение ввода символом \$ (вместо точки с запятой) позволяет вычислить результат введённой команды, но не выводить его на экран.

Правила ввода чисел в Maxima точно такие, как и для многих других подобных программ. Целая и дробная часть десятичных дробей разделяются символом точка. Перед отрицательными числами ставится знак минус. Числитель и знаменатель обыкновенных дробей разделяется при помощи символа / (прямой слэш). Обратите внимание, что если в результате выполнения операции получается некоторое символьное выражение, а необходимо получить конкретное числовое значение в виде десятичной дроби, то решить эту задачу позволит применение флага numer. В частности он позволяет перейти от обыкновенных дробей к десятичным. Преобразование к форме с плавающей точкой осуществляет также функция float.

```
(%i2) 3/15;  
(%o2)  $\frac{1}{5}$   
(%i3) 3/15, numer;  
(%o3) 0.2  
(%i4) float(3/15);  
(%o4) 0.2
```

В Maxima определены арифметические операции + (сложение), – (вычитание или унарный минус как знак отрицательного числа), \* (умножение), / (деление). Существует несколько идентичных способов задания возведения в степень (^, \*\*, ^^). Операция нахождение факториала обозначается восклицательным знаком, например, 5!.

```
(%i5) 202+348*585-2**10/4;
(%o5) 203526
(%i6) 25!;
(%o6) 15511210043330985984000000
(%i7) 2**100;
(%o7) 1267650600228229401496703205376
```

Для увеличения приоритета операции, как и в математике, используются круглые скобки: ().

```
(%i9) 25+5*45;
(%o9) 250
(%i10) (25+5)*45;
(%o10) 1350
```

## Встроенные константы. Переменные и функции

В Maxima для удобства вычислений имеется ряд встроенных констант. Самые распространённые из них - %pi (число пи), %e (число e – основание натуральных логарифмов), %i (мнимая единица). По умолчанию все вычисления проводятся на множестве комплексных чисел. Символ % используется в начале служебных имен Maxima для исключения пересечений с пользовательскими идентификаторами. Одиночный символ % используется для обращения к результату предыдущего вычисления, а одиночный символ подчеркивания – к предыдущему вводу.

```
(%i8) %pi;
(%o8) π
(%i9) 180/%pi, numer;
(%o9) 57.29577951308232
```

Кроме того, все ячейки нумеруются символами %i номер (ячейки ввода) и %o номер (ячейки вывода), что позволяет обращаться из любой ячейки программы к любому ранее вычисленному значению.

```
(%i1) 1/2+1/3;
(%o1)  5
      6
(%i2)  _;
(%o2)  5
      6
(%i3)  %;
(%o3)  5
      6
```

Помимо этого, существует и традиционный способ запоминания вычисленных значений – посредством переменных. Имена переменных и функций (идентификаторы) могут включать в себя буквы латинского алфавита, символ подчеркивания и цифры. Идентификаторы не могут начинаться с цифры. Необходимо также подчеркнуть, что прописные и строчные буквы в идентификаторах различаются, так что имена temp, TEMP и TemP задают разные объекты. Присваивание значения переменной осуществляется с использованием символа : (двоеточие), например, x:5.

```
(%i10) a:5;
(a) 5
(%i11) f:3/a;
(f)  3
     5
(%i12) f,number;
(%o12) 0.6
```

В правой части оператора присваивания могут быть также и уравнения, то есть выражения, содержащие знак равенства. Допустимы арифметические операции над уравнениями:

```
(%i4)  x:y+6=8;
(x)    y+6=8
(%i5)  x+100;
(%o5)  y+106=108
(%i6)  _-108;
(%o6)  x-8
```

Вывод в последнем примере требует пояснения. Дело в том, что если в уравнении правая часть тождественна нулю, то правая часть (вместе со знаком равенства) опускается.

Зарезервированные слова, использование которых в качестве имён переменных вызывает синтаксическую ошибку: integrate, next, from, diff, in, at, limit, sum, for, and, elseif, then, else, do, or, if, unless, product, while, thru, step.

Если необходимо удалить значение переменной (очистить её), то применяется метод kill. Например, kill(x) удаляет переменную x; kill(all) удаляет все введенные ранее переменные. Заметим также, что kill(all) полностью очищает память, в том числе и ранее вычисленные значения. Поэтому после использования kill(all) нумерация ячеек начинается сначала. Рекомендуем использовать kill(all) для разделения задач в одном файле, чтобы избежать ошибочного повторного использования результатов из других задач.

В Maxima имеется достаточно большой набор встроенных математических функций.

Для записи функции необходимо указать ее название, а затем, в круглых скобках записать через запятую значения аргументов.

```
(%i13)  x:48;
(x)    48
(%i14)  y:sqrt(x+1);
(y)    7
```

Основные математические функции представлены в таблице 1:

Таблица 1. Основные математические функции системы Maple.

sin — синус;	asin — арксинус;
cos — косинус;	acos — арккосинус;
tan — тангенс;	atan — арктангенс;
sec — секанс;	asec — арксеканс;
csc — cosecant;	acsc — арккосеканс;
cot — котангенс.	acot — арккотангенс.
sinh — гиперболический синус;	asinh — гиперболический арксинус;
Cosh — гиперболический косинус;	Acosh — гиперболический арккосинус;
tanh — гиперболический тангенс;	atanh — гиперболический арктангенс;
sech — гиперболический секанс;	asech — гиперболический арксеканс;
Csch — гиперболический cosecant;	Acsch — гиперболический арккосеканс;
Coth — гиперболический котангенс;	atanh — гиперболический арккотангенс;
log — натуральный логарифм;	exp — экспонента;
sqrt — квадратный корень	abs — модуль числа;
mod — остаток от деления	sign — знак аргумента
max — максимальное из чисел	min — минимальное из чисел
lcm — наименьшее общее кратное;	gcd — наибольший общий делитель;
cabs — модуль комплексного числа;	carg — аргумент комплексного числа;
realpart — действительная часть комплексного числа;	imagpart — мнимая часть комплексного числа;

Операция divide(a,b) реализует деление с остатком. Результатом является двухэлементный список, первый элемент которого принимает значение неполного частного, а второй — остатка. Как и в функциях нахождения наибольшего общего делителя gcd(a,b) и наименьшего общего кратного lcm(a,b), в качестве аргументов допускается задавать выражения в виде полиномов.

```
(%i1) divide(120,55);
(%o1) [2,10]
(%i2) divide(x^4-1,(x-1)^4);
(%o2) [1,4 x^3-6 x^2+4 x-2]
```