Использование системы компьютерной алгебры Maxima в преподавании математики студентам колледжа

Автор:
Савинова Лариса Николаевна
ГОУ ВО МО "ГГТУ",
Промышленно-экономический колледж
г. Орехово-Зуево

Введение

В настоящее время, информационное образование является необходимой составляющей подготовки специалиста практически в любой области. Поэтому внедрение в педагогическую практику информационных технологий, в частности, пакетов прикладных математических программ, позволит повысить эффективность учебного процесса и, в перспективе, улучшить качество подготовки выпускаемых специалистов.

По новым стандартам образования ФГОС СПО при овладении профессий у студентов должны формироваться информационно-коммуникационные компетенции. Знания умения и навыки по информатике необходимы для изучения других общеобразовательных предметов, для их использования в ходе изучения спец. дисциплин профессионального цикла, в практической деятельности и повседневной жизни.

Современное образование немыслимо без применения компьютерных технологий. Они позволяют моделировать явления и процессы, осваивать теоретический материал, осуществлять контроль уровня знаний и умений студентов. Поэтому целесообразно в теории и практике обучения применять различные виды ИКТ.

В настоящее время роль математических знаний и математических методов исследования постоянно возрастает, поэтому математическое образование имеет особое значение. Важно наряду с изучением собственно математики показать ее прикладные возможности, научить учащихся использовать свои математические знания для решения конкретных задач. Этому очень поможет использование информационных технологий, в частности компьютерных программ.

В настоящей статье рассмотрены основные теоретические и практические вопросы по использованию пакета *Maxima* для решения математических задач.

Maxima – это программа для выполнения математических вычислений, символьных преобразований и построения графиков.

Maxima является универсальным математическим пакетом, позволяющим решать большое количество сложных математических задач без использования программирования. Существуют две идентичные версии пакета для операционных систем Windows и Linux.

Преимуществами программы являются ее возможность свободного использования (распространяется на основе лицензии GNU), широкий класс решаемых задач, в программе есть справка и инструкция по работе с программой.

Среди возможностей Maxima: решение уравнений, построение двухмерных и трехмерных графиков, упрощение выражений, использование широкого спектра математических функций, дифференцирование и интегрирование функций и многое другое. С каждой новой версией в Maxima появляются новые функциональные возможности и виды решаемых задач.

Применение компьютерных программных средств на уроках математики позволяет учителю не только разнообразить традиционные формы обучения, но и решать самые разные задачи: повысить наглядность обучения, обеспечить его дифференциацию, облегчить контроль знаний учащихся, повысить интерес к предмету, познавательную активность студентов.

1. Основные понятия системы компьютерной математики Maxima

Изучая и анализируя современное состояние применения прикладных программных средств (ППС) в процессе обучения, следует отметить, что на сегодняшний день уже сформировался определенный набор компьютерных программ, эффективно используемых в процессе преподавания математического цикла.

Среди современных ППС особое место занимают пакеты прикладных математических программ (ППМП). Универсальные математические пакеты предоставляют новые широкие возможности для совершенствования образования на всех, без исключения, его этапах – от целенаправленного обучения и образования до комплексной подготовки обучаемого к профессиональной деятельности и самореализации. Велика роль пакетов прикладных программ при изучении математики. Облегчая решение сложных задач, они снимают психологический барьер в изучении математики и делают этот процесс интересным и более простым.

Maxima — математическая система символьных и численных вычислений. Программа работает в консольном режиме и виде оконного приложения. При проведении вычислений, Maxima использует точные дроби, целые числа и числа с плавающей точкой, что позволяет проводить вычисления с очень высокой точностью.

История проекта, известного ныне под именем Maxima, началась еще в конце 60-х годов в легендарном MIT (Massachusetts Institute of Technology— Массачусетский Технологический институт), когда в рамках существовавшего в те годы большого проекта MAC началась работа над программой символьных вычислений, которая получила имя Macsyma (от MAC Symbolic MAnipulation). Архитектура системы была разработана к июлю 1968 г., непосредственно программирование началось в июле 1969. В качестве языка для разработки системы был выбран Lisp, и история показала, насколько это был правильный выбор: из существующих в то время языков программирования он единственный продолжает развиваться и сейчас — спустя почти полвека после старта проекта. Macsyma была закрытым коммерческим проектом; его финансировали государственные и частные организации, среди которых были вошедшее в историю ARPA (Advanced Research Projects Agency), Энергетический и Оборонный Департаменты США (Departments of Energy & Defence, DOE and DOD). Проект активно развивался, а организации, контролирующие его, менялись не раз, как это всегда бывает с долгоживущими закрытыми проектами. В 1982 году профессор Уильям Шелтер начал разрабатывать свою версию на основе этого же кода, под названием Maxima. В 1998 году Шелтеру удалось получить от DOE права на публикацию кода по лицензии GPL. Первоначальный проект Macsyma прекратил свое существование в 1999 году. Уильям Шелтер продолжал заниматься разработкой Maxima вплоть до своей смерти в 2001 году. Но, что характерно для открытого ПО, проект не умер вместе со своим автором и куратором. Сейчас проект продолжает активно развиваться, и участие в нем является лучшей визитной карточкой для математиков и программистов всего мира.

Maxima является универсальным математическим пакетом, позволяющим решать большое количество сложных математических задач без использования программирования. Существуют две идентичные версии пакета для ОС Windows и Linux.

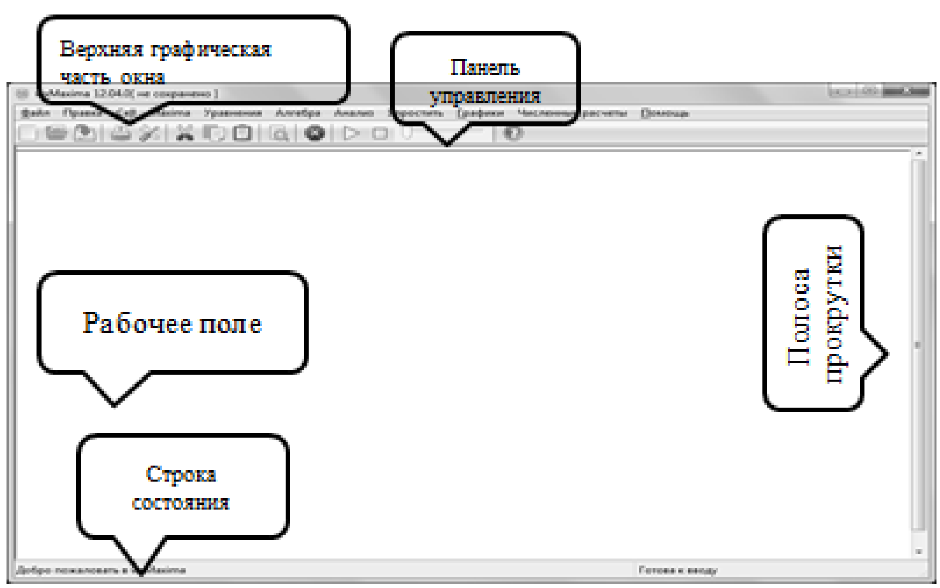
На данный момент ППП *Maxima* может:

– использоваться как обычный калькулятор для простых вычислений;
– вычислять и упрощать символьные выражения;
– использовать для вычисления интегралов и производных функций;
– решать системы линейных алгебраических уравнений, работать с матрицами и определителями;
– решать нелинейные алгебраические уравнения;
– решать системы нелинейных алгебраических уравнений;
– строить графики, как в декартовых, так и в полярных координатах, различные диаграммы и гистограммы;
– решать дифференциальные уравнения;
– создавать документы, которые хорошо выглядят в отчетах.

С каждой новой версией в пакете *Maxima* появляются новые функциональные возможности и виды решаемых задач. Пакет позволяет решать символьные и численные выражения. Имеется довольно обширная литература для работы с пакетом. Кроме того, есть возможность познакомиться с системой Maxima-онлайн – аналогично производя вычисления.

2. Функции и команды программы wxMaxima.

В верхней графической части окна интерфейса wxMaxima демонстрируется версия загруженной программы. Ниже расположена панель управления, содержащая следующие команды: файл, правка, cell, maxima, уравнения, алгебра, анализ, упростить, графики, численные расчеты и помощь. Далее расположено рабочее поле, ниже строка состояния.



В *Maxima* для ввода функций и команд существует два способа. Первый способ: в панели управления выбираем нужную команду и вводим пример. Ввод команд через диалоговые окна упрощает работу с программой для новичков. Второй способ заключется в следующем: ставим курсор в рабочее поле набираем нужную команду или фукцию. Но таким способом мы решаем, если знаем «название» функции или команды. Разделяются функции и команды символом «;» (*точка с запятой*).

При использовании интерфейса *Maxima*, можно выделить в окне вывода результатов необходимую формулу и, вызвав контекстное меню правой кнопкой мыши, скопировать любую формулу в текстовом виде или в виде графического изображения, для последующей вставки в какой-либо документ.

После ввода команды необходимо нажать Enter для ее обработки и вывода результата. Завершение ввода $ (вместо точки с запятой) позволяет вычислить результат введенной команды, но не выводить его на экран. В случае, когда выражение надо отобразить, а не вычислить, перед ним необходимо поставить знак «’» (одинарная кавычка).

Две одинарные кавычки последовательно, примененные к выражению во входной строке, приводят к замещению входной строки результатом вычисления выражения.

Обозначение арифметических операций в Maxima ничем не отличается от классического представления: + , - , * , /. Возведение в степень можно обозначать несколькими способами: ˆ , **. Извлечение корня степени n записываем, как степень 1/n. Операция нахождение факториала обозначается восклицательным знаком, например 5!. Для увеличения приоритета операции, как и в математике, используются круглые скобки: ().

Список основных арифметических и логических операторов приведен в таблицах ниже.

Таблица 1. Арифметические операторы

Обозначение	Оператор
+	оператор сложения
-	оператор вычитания или изменения знака
/	оператор деления
*	оператор умножения
ˆ или **	оператор возведения в степень

Таблица 2. Логические операторы

Обозначение	Оператор
<	оператор сравнения меньше
>	оператор сравнения больше
<=	оператор сравнения меньше или равно
>=	оператор сравнения больше или равно
#	оператор сравнения не равно
=	оператор сравнения равно
and	логический оператор и
or	логический оператор или
not	логический оператор не

В Maxima для удобства вычислений имеется ряд встроенных констант. Самые распространенные из них показаны в следующей таблице 3.

Таблица 3. Основные константы

Название	Обозначение Maxima
слева (в отношении пределов)	minus
справа (в отношении пределов)	plus
минус бесконечность	minf
плюс бесконечность	inf
e (экспонента)	%e
число π	%pi
Мнимая единица $\sqrt{-1}$	%i
Истина	true
Ложь	false
Золотое сечение $\frac{1+\sqrt{5}}{2}$	%phi

Для хранения результатов промежуточных расчетов применяются переменные. Заметим, что при вводе названий переменных, функций и констант важен регистр букв, так переменные x и X - две разные переменные. Присваивание значения переменной осуществляется с использованием символа «:» (двоеточие), например x:5.

Если необходимо удалить значение переменной (очистить ее), то применяется метод kill:

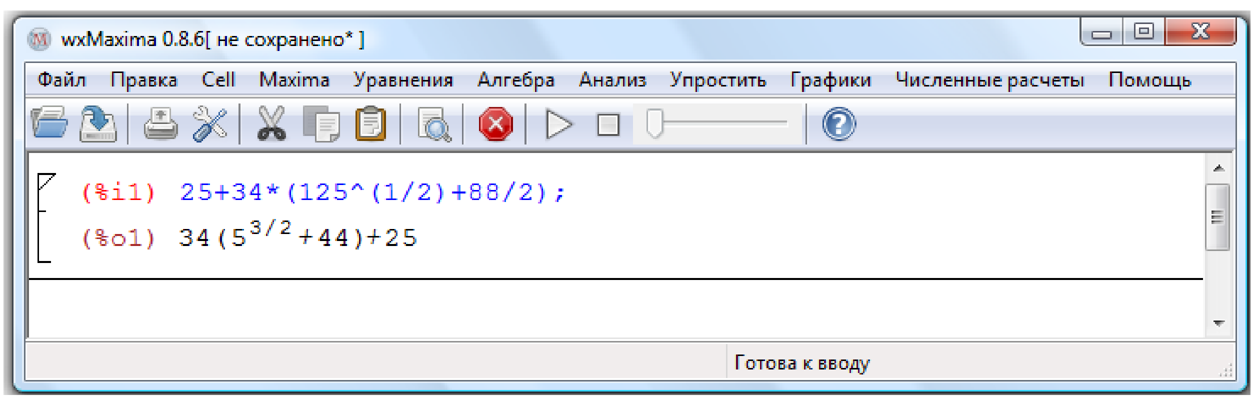
kill(x) - удалить значение переменной x;

kill(all) - удалить значения всех используемых ранее переменных.

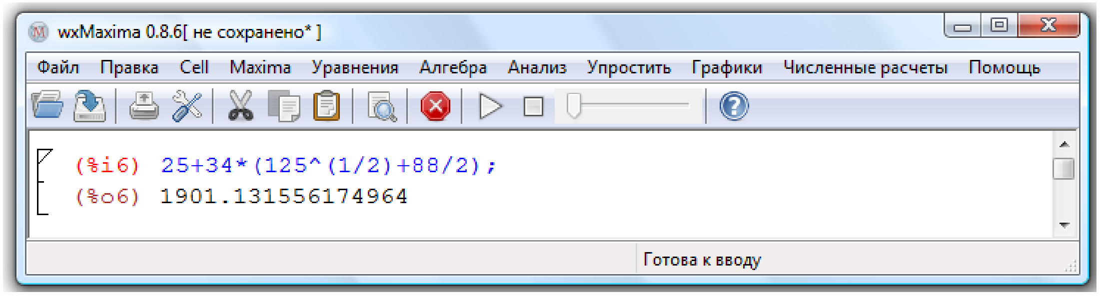
Зарезервированные слова, использование которых в качестве имен переменных вызывает синтаксическую ошибку:

Integrate, nex,t from, diff, in, at, limit, sum, for, and, elseif, then, else, do, or, if, unless, product, while, thru, step.

Пример. Арифметические выражения вводятся в поле ввода.



После нажатия <Ctrl>+<R> получаем:



Если необходимо вычислить дробные выражения, то системы сама может привести выражения к общему знаменателю

Например,

```
(%i18) 3/7+5/9-1/5;
        247
(%o18)  ---
        315
```

Если значение нужно получить в виде десятичной дроби, то незабудьте воспользоваться панелью ЧИСЛЕННЫЕ РАСЧЕТЫ выбрав команду float

```
(%i18) 3/7+5/9-1/5;
        247
(%o18)  ---
        315
```

```
(%i19) float(%), numer;
(%o19) 0.78412698412698
```

Пример. Вычислить $\dfrac{\frac{1}{2}+\frac{3}{5}-\frac{7}{11}}{\frac{1}{3}+\frac{4}{5}}*10$

```
(%i58) (1/2+3/5-7/11)/(1/3+4/5)*10;
        45
(%o58)  --
        11
```

Таблица 4.

Некоторые встроенные математически функции системы Maxima

Abs(x)	Модуль числа x
Sqrt(x)	Квадратный корень из x
Acos(x)	Арккосинус аргумента x
Acot(x)	Арккотангенс аргумента x
Asin(x)	Арксинус аргумента x
Atan(x)	Арктангенс аргумента x
Sin(x)	Синус аргумента x
Tan(x)	Тангенс аргумента x
Log(x)	Натуральный логарифм x
Exp(x)	Экспонента x

3. Решение задач элементарной математики и задач теории чисел

3.1. Преобразование алгебраических выражений

В системе Maxima имеется множество возможностей для преобразования выражений. Основной функцией, обрабатывающей выражения является функция ***ev***. Ее синтаксис

ev(выражение);

ev(выражение, flag1,flag2,…);

ev(выражение, x=a+b,y:c/d,…);

ev(выражение,flag1, x=a+b,flag2, y:c/d, …);

имя функция можно опускать. Примеры использования данной функции можно так же посмотреть в справочной системе.

По умолчанию в системе Maxima активна функция упрощения, однако эта функция не всегда способна упростить выражение. В дополнение к ней имеется целый ряд команд, которые работают с рациональными иррациональными, тригонометрическими и логарифмическими выражениями.

Функция ***factor*** представляет выражение в виде произведения нескольких множителей.

Пример. Разложить на множители $x^4+3x^3+2x^2+x+1$

```
(%i85) factor(x^4+3*x^3+2*x^2+x+1);
```

(%o85) $(x+1)(x^3+2x^2+1)$

Функция f***actorsum*** раскладывает на множители отдельные слагаемые, однако надеяться на эту функцию особенно не стоит, поскольку она не всегда видит какие возможны преобразования.

Функция выражение and раскрывает скобки

Пример. Раскрыть скобки $(x+1)(x^2+3)(x^3+7)$

```
(%i2) expand((x+1)*(x^2+3)*(x^3+7));
```

(%o2) $x^6+x^5+3x^4+10x^3+7x^2+21x+21$

Функция ***combine*** объединяет слагаемые с одинаковыми знаменателями.

Функция ***xthru*** приводит выражение к общему знаменателю, не раскрывая скабок и не раскладывая на множители слагаемые.

Пример. Привести к общему знаменателю выражение $\dfrac{x^3+1}{x^2+1}+\dfrac{2x+3}{x^2-1}$

```
(%i10) xthru((x^3+1)/(x^2+1)+(2*x+3)/(x^2-1));
```

(%o10) $\dfrac{(x^2-1)(x^3+1)+(2x+3)(x^2+1)}{(x^2-1)(x^2+1)}$

Функция *multthru* умножает каждое слагаемое суммы на множители, причем при умножении скобки в выражении не раскрываются.

```
(%i11) example(multthru);
(%i12) (-f(x))/(x-y)^3-1/(x-y)+x/(x-y)^2
```

(%o12) $-\frac{1}{x-y}+\frac{x}{(x-y)^2}-\frac{f(x)}{(x-y)^3}$

```
(%i13) multthru((x-y)^3,%)
```

(%o13) $x\,(x-y)-(x-y)^2-f(x)$

```
(%i14) ratexpand(%)
```

(%o14) $-y^2+x\,y-f(x)$

```
(%i15) ((a*b)^2+2*a*b*s+(b+a)^10*s^2)/(a*b*s^2)
```

(%o15) $\frac{(b+a)^{10}\,s^2+2\,a\,b\,s+a^2\,b^2}{a\,b\,s^2}$

```
(%i16) multthru(%)
```

(%o16) $\frac{2}{s}+\frac{a\,b}{s^2}+\frac{(b+a)^{10}}{a\,b}$

```
(%i17) multthru(a . (f+c . (e+d)+b))
(%o17) a . f+a . c . (e+d)+a . b
(%o17) done
```

Функция ***rat*** — преобразовывает рациональное выражение к канонической форме: раскрывает все скобки, затем приводит все к общему знаменателю, суммирует и сокращает; приводит все числа в конечной десятичной записи к рациональным. Каноническая форма автоматически «отменяется» в случае любых преобразований, не являющихся рациональными

Функция ***ratsimp*** — упрощает выражение за счет рациональных преобразований. Работает в том числе и «вглубь», то есть иррациональные части выражения упрощаются, в том числе, и все рациональные элементы внутри них.

Фунция ***fullratsimp*** — функция упрощения рационального выражения методом последовательного применения к переданному выражению функции ratsimp(). За счет этого функция работает несколько медленнее, чем ratsimp(), зато дает более надежный результат.

Функция ***radcan*** — функция упрощения логарифмических, экспоненциальных функций и степенных с нецелыми рациональными показателями, то есть корней (радикалов).

Функция ***rootscotract*** упрощает возведения в степень.

Функция ***logcotract*** упрощает логарифмические выражения

Пример. Разложить на множители функции $x^3+x^2z+xyz+y^2z-y^3$ и $x^5+x^4y+2y^5-x^2y^3-xy^4-2x^3y^2$

```
(%i23) factor(x^3+x^2*z+x*y*z+y^2*z-y^3);
```

(%o23) $(y^2+x\,y+x^2)(z-y+x)$

```
(%i24) factor(x^5+x^4*y+2*y^5-x^2*y^3-x*y^4-2*x^3*y^2);
```

(%o24) $(y-x)^2\,(2\,y+x)(y^2+x\,y+x^2)$

Пример. Упростить выражение $\left(x^2 - xy + y^2\right)^3 + \left(x^2 + xy + y^2\right)^3$

```
(%i28) ratsimp((x^2-xy+y^2)^3+(x^2+xy+y^2)^3);
```
(%o28) $2\,y^6+6\,x^2\,y^4+(6\,xy^2+6\,x^4)\,y^2+6\,x^2\,xy^2+2\,x^6$

```
(%i32) factor(%);
```
(%o32) $2\,(y^2+x^2)(y^4+2\,x^2\,y^2+3\,xy^2+x^4)$

Пример. Разложить выражение на простейшие дроби $\dfrac{x^3 + 4x^2 - 5x + 6}{(x-2)^2(x-1)(x^2+1)}$

Выполнить это задание можно используя меню АНАЛИЗ выбрав команду partfrac

```
(%i56) partfrac((x^3+4*x^2-5*x+6)/((x-2)^2*(x-1)*(x^2+1)), x);
```
(%o56) $-\dfrac{2\,x+4}{5\,(x^2+1)}+\dfrac{3}{x-1}-\dfrac{13}{5\,(x-2)}+\dfrac{4}{(x-2)^2}$

Для преобразования тригонометрических функций имеются специальные функции.

Функции *trigexpand* раскладывает все тригонометрические функции от сумм в суммы произведений тригонометрических функций.

Функция *trigreduce* представляет все произведения тригонометрических функций в тригонометрические функции от сумм. Функция не всегда работает до конца, поэтому полезен, бывает ее повторный вызов.

Функция *trigsimp* применяет к любому тригонометрическому выражению основное тригонометрическое тождество

Функция *trigrat* упрощает выражение с тригонометрическими функциями, но бывает, что работа этой функции занимает очень долго времени

```
(%i17) trigsimp((cos(x)^2-sin(x)^2)*sin(y)+2*cos(x)*sin(x)*cos(y));
```
(%o17) $\left(2\cos(x)^2-1\right)\sin(y)+2\cos(x)\sin(x)\cos(y)$

```
(%i18) trigrat(%);
```
(%o18) $\sin(y+2\,x)$

3.2. Решение некоторых задач теории чисел

В системе имеется пакет, позволяющий решать задачи теории чисел NumberTheory. Рассмотрим некоторые функции этого пакета

Функция: ***cf(выражение)*** Преобразует *выражение* в непрерывную дробь. Выражения могут быть цепными дробями, целыми или рациональными числами. Maxima не знает об операциях на непрерывных дробей вне функции cf .

cfоценивает свои аргументы после связывания listarith к false.

cf создает дробь, в виде списка.

Пример. Вычислить $\left(2+\cfrac{1}{4+\cfrac{1}{7}}\right)\cdot\left(13+\cfrac{1}{11+\cfrac{1}{9}}\right)+\left(2+\cfrac{1}{7}\right):\left(5+\cfrac{1}{3+\cfrac{1}{2}}\right)$

Набираем в командной строке команду cf и соответствующие дроби в виде списков получаем:

```
(%i10)  cf ([2, 4, 7]*[13, 11, 9] + [2, 7]/[5, 3, 2]);
(%o10) [29,1,2,1,11,1,9,2,1,13]
```

Функция: **cfdisrep***(список)* представляет список [a,b,c,...]. в виде непрерывной дроби

$$a+\cfrac{1}{b+\cfrac{1}{c+\ldots}}$$

```
(%i18) cf([2,3,5,17,8]);
(%o18) [2,3,5,17,8]
(%i19) cfdisrep (%);
```

$$\text{(\%o19)}\quad 2+\cfrac{1}{3+\cfrac{1}{5+\cfrac{1}{17+\cfrac{1}{8}}}}$$

Пример. Вычислить и представить результат в виде непрерывной дроби

$$\left(1+\cfrac{1}{5+\cfrac{1}{(-7)}}\right)+\left(2+\cfrac{1}{-3+\cfrac{1}{4}}\right)$$

Вводим дробь в виде списка, а затем с помощью команды cfdisrep список преобразуем в непрерывную дробь:

```
(%i2) cf ([1, 5, -7] + [2, -3, 4]);
(%o2) [2,1,5,2,1,19]
(%i3)  cfdisrep (%);
```

$$\text{(\%o3)}\quad 2+\cfrac{1}{1+\cfrac{1}{5+\cfrac{1}{2+\cfrac{1}{1+\cfrac{1}{19}}}}}$$

Функция **cflength** представляет обычное арифметическое выражение вида $a+\cfrac{1}{b+\cfrac{1}{c+\ldots}}$ списком [a,b,c,...].

Пример. Представить в виде списка дробь $2+\cfrac{1}{3+\cfrac{1}{5+\cfrac{1}{4}}}$

```
(%i28) cflength: 1$
        cf (2+(1/(3+1/(5+1/4))));
(%i29)
(%o29) [2,3,5,4]
```

Функция: **divsum***(n)*вычисляет сумму делителей *n.*

Пример: Вычислите сумму делителей чисел 12, 11, 210.

В командной строке вводим команду **divsum** и соответствующее число на экране получаем результат:

```
(%i4) divsum (12);
(%o4) 28

(%i5) divsum (11);
(%o5) 12

(%i6) divsum (210);
(%o6) 576
```

Функция: **factorial** *(x)* Представляет функции факториала. Maxima относится factorial (*x*) так же, как *x!*

Функция: **fib** *(n*Вычисляет числа Фибоначчи. fib(0) равна 0 и fib(1) равны 1, и fib (- *n*) равна $(-1)^{n+1} fib(n)$.

Функция: **ifactors***(n)*Для натурального *n* раскладывает на простые множители натуральные числа

Пример. Разложить на множители число 515753196

Вводим нужную команду, и на экране появляется результат

```
(%i17) ifactors(515753196);
(%o17) [[2,2],[3,1],[7,1],[23,1],[266953,1]]
```

Таким образом, $515753196 = 2^2 \cdot 3 \cdot 7 \cdot 23 \cdot 266953$.

Пример. Разложить на простые множители 25!

```
(%i26) ifactors(25!);
(%o26) [[2,22],[3,10],[5,6],[7,3],[11,2],[13,1],[17,1],[19,1],[23,1]]
```

Таким образом, $25! = 2^{22} \cdot 3^{10} \cdot 5^6 \cdot 7^3 \cdot 11^2 \cdot 13 \cdot 17 \cdot 19 \cdot 23$

Функция: **lcm***(выражение 1 , ..., выражение_n)*. Вычисляет наименьшее общее кратное своих аргументов. Аргументы могут быть многочленами, а также целыми числами.

Активировать функцию можно после введения команды load ("functs")

Функция: **next_prime***(n)* Находит наименьшее простое число большее, чем *n.*

Функция: **partfrac***(выражение , var)*раскладывает выражение на простейшие дроби по заданной переменной.

Функция: **power_mod***(a , n , m).*Использование модульного алгоритма вычисления $a^n \pmod{m}$, где a и n целые числа и *m* целое положительное число.

Функция: **primep***(n)* Определяет простое или составное числа n Если составное, то на экране результат false, а если на экране true , *n* простое число с очень высокой вероятностью.

Функция: **prev_prime***(n)* Находит наибольшее простое число меньшее, чем *n.*

Функция: **totient***(n)*Находит количество целых чисел меньше или равных *n,* взаимно простых с *n.* (вычисляет функцию Эйлера)

Пример. Вычислить функцию Эйлера для чисел 5 и 375

```
(%i44) totient (5);
(%o44) 4
```

```
(%i45) totient (375);
(%o45) 200
```

Пример. Сколько чисел в интервале от 1 до 1200 не взаимно простых с 30?

Всего чисел взаимно простых с 1200 и меньших его, будет

```
(%i51) totient (1200);
(%o51) 320
```

Все остальные $1200-320=880$ чисел будут с числом $30=2\cdot 3\cdot 5$

```
(%i66) ifactors(30);
(%o66) [[2,1],[3,1],[5,1]]
```

иметь общие делители, большие единицы.

3.3. Работа с комплексными числами.

Комплексное число в Maxima определено в алгебраической форме с помощью сложения действительной части выражения и произведения %i (мнимой единицы) и мнимой части.

Преобразование комплексных выражений может осуществляться функциями для работы с алгебраическими выражениями (***radcan***, ***expand*** и др.), но предусмотрен и ряд специфических функций, рассчитанных на операции именно с комплексными числами.

При упрощении частных, корней, и других функций комплексных выражений обычно используются функций ***realpart***, ***imagpart***, ***rectform***, ***polarform***, ***abs***, ***carg***.

Вычисление модуля комплексного числа осуществляется функцией ***cabs***. Аргумент комплексного выражения вычисляется при помощи функции ***carg***. Комплексный аргумент - θ в пределах [−π, π] таким образом, что r exp (тета %i) = z где r - модуль комплексного числа z. Следует учитывать, что ***carg*** - вычислительная функция, не предназначенная для упрощения комплексных выражений.

Пример. Вычислить модуль и аргумент комплексных чисел а)12; б) -4; с) (2+3i).

Вычислим модуль данных чисел:

```
(%i10) cabs(12);
(%o10) 12

(%i11) cabs(-4);
(%o11) 4

(%i12) cabs(2+%i*3);
```
(%o12) $\sqrt{13}$

А теперь вычислим аргумент этих чисел:

```
(%i7) carg(12);
(%o7) 0

(%i8) carg(-4);
```
(%o8) π

```
(%i9) carg(2+%i*3);
```
(%o9) $\operatorname{atan}\left(\frac{3}{2}\right)$

Для преобразования комплексных выражений используют также функцию ***demoivre***.

Когда переменная demoivre установлена (**demoivre**=**true**), комплексные показательные функции преобразованы в эквивалентные выражения в терминах тригонометрических функций: exp (a+b*%i) упрощает к виду %ea∗(cos(b) + %i ∗sin(b)) если выражение b не содержит %i. Значение по умолчанию ***demoivre*** - ***false***. Кроме того, преобразование различных форм комплексных чисел осуществляется функцией **exponentialize**, которая преобразует тригонометрические и гиперболические функции в экспоненциальную форму.

Пример.

```
(%i22) demoivre:true;
       demoivre (exp (3+3/2 * %pi * %i));
        demoivre (exp (%pi+3/2 * %pi * %i));

(%o22) true
(%i23)
```
(%o23) $-\%e^{3}\ \%i$

(%i24)

(%o24) $-\%e^{\pi}\ \%i$

Комплексно-сопряжённые выражения вычисляются при помощи функции ***conjugate*** (x).

Пример. Запишите число сопряженное данному а) 2+34i; б) -5i

```
(%i38) conjugate (2 + 3*%i);
(%o38) 2-3 %i

(%i39) conjugate (-5*%i);
(%o39) 5 %i
```

4. Решение задач математического анализа

При решении задач математического анализа с помощью системы Maxima используется меню АНАЛИЗ. Основные команды этого меню представлены на рисунке.

Рисунок. Меню АНАЛИЗ

4.1. Вычисление пределов числовых последовательностей

Функция ***limit(f(x), x, a)***; в тех случаях когда левый и правый пределы не совпадают можно уточнить какой именно предел вычисляется. Существует четыре специальных значения

inf - $+\infty$

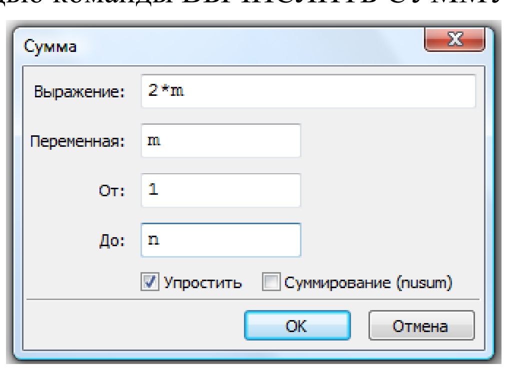
minf - $-\infty$

und - $\pm\infty$

ind - неопределенность

Пример. Вычислить предел $\lim\limits_{n\to\infty}\dfrac{2+4+6+...+2n}{7n^2}$

Преобразуем выражение находящееся под знаком предела. Вычислим сумму числителя с помощью команды ВЫЧИСЛИТЬ СУММУ меню АНАЛИЗ

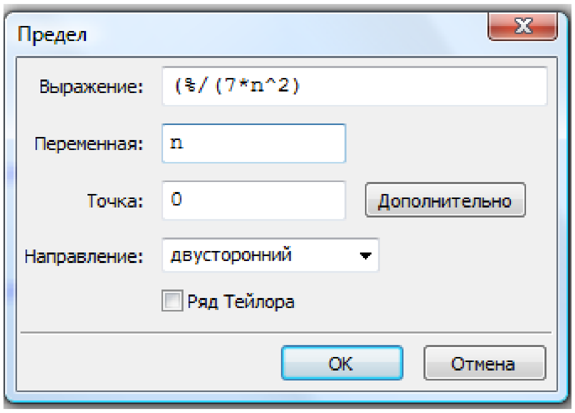


После чего на экране появляется

```
(%i5) sum(2*m, m, 1, n), simpsum;
```

(%o5) n^2+n

Используя полученный результат вычисляем предел использую меню АНАЛИЗ команду ВЫЧИСЛИТЬ ПРЕДЕЛ



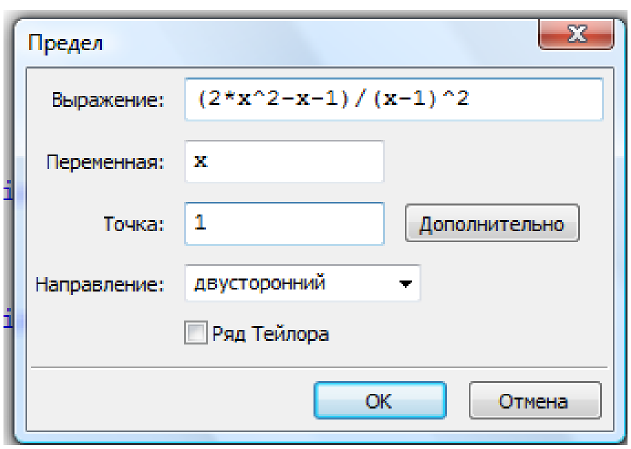
На экране появится

```
(%i6) limit(%/(7*n^2), n, inf);
```

(%o6) $\frac{1}{7}$

Пример. Вычислить предел $\lim\limits_{x\to 1}\dfrac{2x^2-x-1}{(x-1)^2}$

Заполняем диалоговое окно команды ВЫЧИСЛИТЬ ПРЕДЕЛ меню АНАЛИЗ



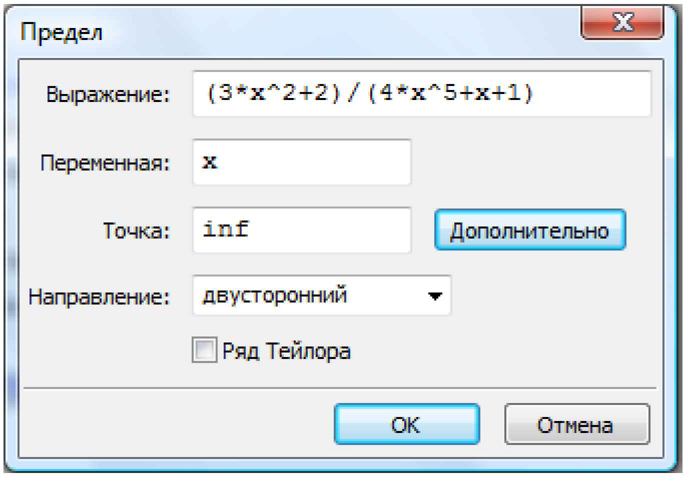
На экране получаем

```
(%i7) limit((2*x^2-x-1)/(x-1)^2, x, 1);
(%o7) infinity
```

Ответ $\lim\limits_{x\to 1}\dfrac{2x^2-x-1}{(x-1)^2}=\infty$

Пример. Вычислить предел $\lim\limits_{x\to\infty}\dfrac{3x^2+2}{4x^5+x+1}$.

Заполняем форму в диалоговом окне



Получаем на экране ответ

```
(%i9) limit((3*x^2+2)/(4*x^5+x+1), x, inf);
(%o9) 0
```

Как видим раскрытие неопределенностей $\left[\dfrac{0}{0}\right]$ и $\left[\dfrac{\infty}{\infty}\right]$ для Maximaне составляет труда.

Пример. Вычислить предел $\lim\limits_{x\to\infty}\left(\sqrt{x^2+1}-\sqrt{x^2-1}\right)$

Заполняем диалоговое окно. В итоге на экране получаем

```
(%i18) limit(((x^2+1)^(1/2))-((x^2-1)^(1/2)), x, inf);
(%o18) 0
```

Без труда Maximaсправляется и с вычислением пределов на основе первого замечательного предела $\lim\limits_{x\to 0}\dfrac{\sin x}{x}=1$

Пример. Вычислить предел $\lim\limits_{x\to\infty} x\sin(\dfrac{1}{x})$

```
(%i19) limit(x*sin(1/x), x, inf);
(%o19) 1
```

Пример. Вычислить предел $\lim\limits_{x\to 1}\left(\dfrac{2}{1-x}-\dfrac{5}{1-x^3}\right)$

Точка x=1 является точкой разрыва для данной функции, поэтому необходимо вычислять право и левосторонние пределы.

В меню АНАЛИЗ выбираем вычисление пределов и в диалоговом окне выбираем НАПРАВЛЕНИЕ заполняем диалоговое окно, на экране получаем

```
(%i55) limit((2/(1-x))-(5/(1-x^3)), x, 1, minus);
(%o55) ∞

(%i56) limit((2/(1-x))-(5/(1-x^3)), x, 0, plus);
(%o56) -3
```

При операциях с рациональными и ирррациоанальными дробями целесообразно использовать упрощение выражений, разложение на множители т.п., покольку система может допускать ошибки.

Рассмотрим примеры нахождения некоторых пределов с использованием второго замечательного предела.

Пример. Найти предел $\lim\limits_{x\to\infty}\left(\frac{2x+3}{2x-1}\right)^x$

```
(%i4) limit(((2*x+3)/(2*x-1))^x, x, inf);
(%o4) %e^2
```

4.2. Дифференцирование функций одной переменной.

Пакет Maxima предоставляет мощные средства для дифференцирования функций и вычисления дифференциалов. Для вычисления простейшей производной следует в командном окне Maxima ввести команду следующего вида: ***diff(<функция>,<переменная>)***; где <функция> – выражение, задающее функцию (не обязательно одной переменной); <переменная> – имя переменной, по которой будет вестись дифференцирование.

Существует четыре типа вызова этой функции

diff(выражение)

diff(выражение,x)

diff(выражение,x,n)

diff(выражение,x_1,n_1,…,x_m,n_m)

Рассмотрим примеры применения этой функции.

Пример. Найти производную функции $y = x\ln^2 x$

```
(%i11) diff(x*(log(x))^2,x);
(%o11) log(x)^2 +2 log(x)
```

Пример: Найти производную функции $y = e^x arctge^x - \ln\sqrt{1+e^{2x}}$

```
(%i26) diff(((exp)^x*atan((exp)^x)-log(sqrt(1+(exp)^2*x))),x,1);
```

(%o26) $exp^x \log(exp)\,atan(exp^x)+\frac{exp^{2x}\log(exp)}{exp^{2x}+1}-\frac{exp^2}{2(x\,exp^2+1)}$

Если указать ***апостроф*** перед символом ***diff***, то производная не вычисляется и упрощение, обычно предусмотренное по умолчанию, не осуществляется.

4.3. Решение задач на исследование функции

Рассмотрим несколько задач на исследование функции

Пример: Исследовать на наличие экстремума следующую функцию

$$y = \sqrt{1 - \ln^2(x)}$$

Задаём исследуемую функцию

```
(%i3) f(x):=(1-(log(x))^2)^(1/2);
```

$$\text{(\%o3)}\quad f(x):=\left(1-\log(x)^2\right)^{1/2}$$

Производную в форме функции определяем явно, используя функцию **define**

```
(%i16) define(df(x),diff(f(x),x));
```

$$\text{(\%o16)}\quad df(x):=-\frac{\log(x)}{x\sqrt{1-\log(x)^2}}$$

Решая уравнение df(x) = 0 (т.е. f'(x) = 0, находим критические точки

```
(%i17) solve(df(x)=0,x);
(%o17) [x=1]
```

В данном случае критическая точка одна - x = 1.

При работе с системами компьютерной математики удобнее второе достаточное условие экстремума:

Если первая производная f'(x) дважды дифференцируемой функции $y = f(x)$ *равна нулю в некоторой точке* x_0*, а вторая производная в этой точке* $f''(x_0)$ *положительна, то* x_0 *есть точка максимума функции* $y = f(x)$*; если* $f''(x_0)$ *отрицательна, то* x_0 *– точка максимума.*

Пусть $f'(x_0) = 0$, *а* $f''(x_0) > 0$. *Это значит, что* $f''(x\ ) = (f'(x))' > 0$ *также и в некоторой окрестности точки* x_0, *т.е.* $f'(x)$ *возрастает на некотором интервале (a, b), содержащем точку* x_0.

Но $f'(x) = 0$, *следовательно, на интервале (a,* x_0*)* $f'(x) < 0$, *а на интервале (*x_0*, b)* $f'(x) > 0$, *т.е.* $f'(x)$ *при переходе через точку* x_0 *меняет знак с минуса на плюс, т.е.* x_0 *– точка минимума.*

Аналогично рассматривается случай $f'(x_0) = 0$ *и* $f''(x_0) < 0$*.)*

Продолжим исследование функции

Как установлено выше, имеется одна критическая точка: x = 1.

Задаёмся функцией d2f(x)

```
(%i27) define(d2f(x),diff(df(x),x));
```

$$\text{(\%o27)}\quad d2f(x):=\frac{\log(x)}{x^2\sqrt{1-\log(x)^2}}-\frac{1}{x^2\sqrt{1-\log(x)^2}}-\frac{\log(x)^2}{x^2\left(1-\log(x)^2\right)^{3/2}}$$

Вычисляем значение второй производной в критической точке:

```
(%i28) map(d2f,%o17);
```

$$(\%o28)\ [\frac{\log(x)}{x^2\sqrt{1-\log(x)^2}}-\frac{1}{x^2\sqrt{1-\log(x)^2}}-\frac{\log(x)^2}{x^2(1-\log(x)^2)^{3/2}}=-1]$$

Точка x = 1, является максимумом исследуемой функции, т.к. вторая производная в ней оказалась отрицательной. Следует обратить внимание на способ вычисления - функция d2f(x) применяется ко всем элементам списка, полученного при решении уравнения f'(x) = 0 (используется встроенная функция Maxima **map**).

Нахождение наибольших и наименьших значений функции

Наибольшее или наименьшее значение функции на некотором отрезке может достигаться как в точках экстремума, так и в точках на концах отрезка.

Пример. Найти наибольшее и наименьшее значения функции $y=(x-2)^2e^{-x}$ − 6x на отрезке [0,5].

Находим критические точки исследуемой функции

```
(%i27) f(x):=((x-2)^2)*exp^(-x);
```

$$(\%o27)\ f(x):=(x-2)^2\,exp^{-x}$$

```
(%i28) define(df(x),diff(f(x),x));
```

$$(\%o28)\ df(x):=\frac{2(x-2)}{exp^x}-\frac{(x-2)^2\log(exp)}{exp^x}$$

```
(%i29) solve(df(x)=0,x);
```

$$(\%o29)\ [x=\frac{2\log(exp)+2}{\log(exp)},x=2]$$

Результат расчёта - список, включающий два элемента элемент. Причем первый элемент система до конца не посчитала, в данномслучае посчитать самостоятельно не составит труда, можно попробовать и преобразовать с помощью системы.

Создаём новый список, включающий граничные значений и критические точки:

```
(%i30) L:[%o29[2],x=4,x=0,x=5];
```

$$(\%o30)\ [x=2,x=4,x=0,x=5]$$

Применяем функцию f(x) к каждому элементу списка L:

```
(%i31) map(f,L);
```

$$(\%o31)\ [(x-2)^2\,exp^{-x=-2}=0,(x-2)^2\,exp^{-x=-4}=4\,exp^{-x=-4},(x-2)^2\,exp^{-x=0}=4\,exp^{-x=0},(x-2)^2\,exp^{-x=-5}=9\,exp^{-x=-5}]$$

Результат - наибольшие и наименьшие значения - находим в списке полученных значений.

Пример. Исследовать на выпуклость (вогнутость) функцию $y=x(x-1)^3$

Находим производную

(%i1) diff(x*(x-1)^3,x,1);

(%o1) $3(x-1)^2\,x+(x-1)^3$

Ее вторая производная

(%i2) diff(x*(x-1)^3,x,2);

(%o2) $6(x-1)\,x+6(x-1)^2$

(%i12) solve(6*(x-1)*x+6*(x-1)^2=0,x);

(%o12) $[x=\frac{1}{2}, x=1]$

Вторая производная равна нулю при $x=\dfrac{1}{2}$ и $x=1$, $y''>0$ на интервалах $\left(-\infty;\dfrac{1}{2}\right)$ и $(1,+\infty)$ и на этих интервалах функция выпукла вниз; $y''<0$ на интервале $\left(\dfrac{1}{2};1\right)$, следовательно на нем функция выпукла вверх, поэтому $x=\dfrac{1}{2}$ и $x=1$ точки перегиба.

4.4. Интегрирование функций одной переменной.

Неопределенный интеграл вычисляется с помощью команды ***integrate (выражение, x)***. Для вычисления определенного интеграла используется команда ***integrate(выражение,x,a,b***)0

integrate((1+cos(x))ˆ2, x, 0, %pi);

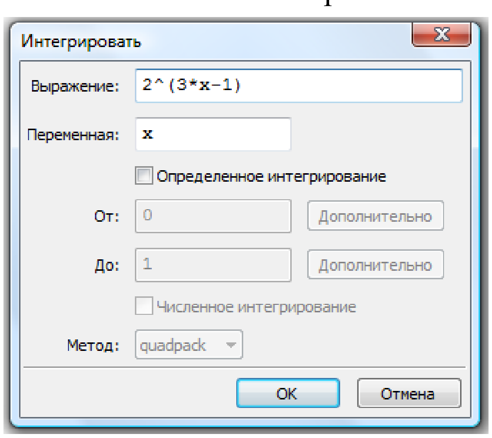
Несобственные интегралы с бесконечными пределами интегрирования вычисляются, если в параметрах команды ***integrate*** указывать, например, ***x***, ***0***, ***inf***.

Численное интегрирование выполняется функцией ***romberg*** или при помощи функций пакета ***quadpack***.

Если требуется вычислить интеграл, зависящий от параметра, то его значение может зависеть от знака этого параметра или каких-либо других ограничений.

Пример. Найти интеграл $\int 2^{3x-1}\,dx$

В меню АНАЛИЗ выбираем ИНТЕГРИРОВАНИЕ в диалоговое окно вводим функцию



На экране получаем:

```
(%i1) integrate(2^(3*x-1), x);
```

$$(\%o1)\quad \frac{2^{3\,x-1}}{3\log(2)}$$

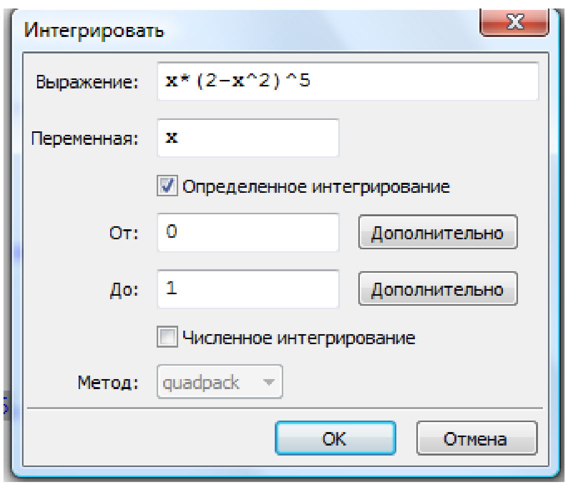
Пример. Найти интеграл $\int \frac{dx}{9x^2-1}$

```
(%i2) integrate(1/(9*x^2-1), x);
```

$$(\%o2)\quad \frac{\log(3\,x-1)}{6}-\frac{\log(3\,x+1)}{6}$$

Пример. Найти интеграл $\int_0^1 x(2-x^2)^5\,dx$

В меню АНАЛИЗ выбираем ИНТЕГРИРОВАНИЕ в диалоговое окно вводим функцию, затем выбираем определённое интегрирование задаем пределы интегрирования



После чего на экране появится результат:

```
(%i4) integrate(x*(2-x^2)^5, x, 0, 1);
```

$$(\%o4)\quad \frac{21}{4}$$

Однако далеко не все интегралы так просто вычисляются в системе Maxima. В некоторых случаях результат получается в неупрощенном виде. В этом случае необходимо применить метод замены переменной в подынтегральном выражении. В Maxima имеется функция, предназначенных для выполнения расчетов шаг за шагом, осуществляющая замену переменной ***changevar***.

Формулу интегрирования по частям:

$$\int u(x)v'(x)dx = u(x)v(x) - \int u'(x)v(x)dx$$

чаще всего придётся применять вручную. В Maxima (в отличие от, например, Maple), функция интегрирования по частям не выделена явно, хотя в отдельных случаях этот способ используется ***integrate***.

Пример. Найти интеграл $\int x^3 e^{x^2} dx$

```
(%i10) integrate(x^3*exp^x^2, x);
```

(%o10) $\frac{\%e^{x^2 \log(exp)} (x^2 \log(exp)-1)}{2 \log(exp)^2}$

Как видим, система не совсем справилась с задачей, и после не сложных преобразований вручную, придем к окончательному ответу.

$$\int x^3 e^{x^2} dx = \frac{x^2 e^{x^2} - e^{x^2}}{2} + C$$

Для вычисления первообразных дифференциальных выражений используется пакет "***antid***"(основные функции пакета - ***antidiff*** и ***antid***). Функция ***antidiff*** выполняет интегрирование выражений с произвольными функциями (в том числе неопределёнными), перед ее первым вызовом следует загрузить пакет (***antid*** отличается от неё форматом выводимого результата).

Если в интеграле требуется сделать замену переменных, используется функция ***changevar***. *Синтаксис вызова* этой функции: ***changevar (выражение, f(x,y), y, x)***.

Функция осуществляет замену переменной в соответствии с уравнением $f(x,y)=0$ во всех интегралах, встречающихся в выражении ***выражение*** (предполачается, что ***y*** - новая переменная, ***x*** - исходная).

При использовании совместно с ***changevar*** часто используется отложенное вычисление интеграла (*одинарная кавычка перед функцией**integrate***).

Интегралы от тригонометрических и логарифмических функций Maxima вычисляет довольно успешно.

Пример. Найти интеграл $\int \sin(3x)\cos(5x)dx$

```
(%i9) integrate(sin(3*x)*cos(5*x), x);
```

(%o9) $-\frac{\cos(8\,x)-4\cos(2\,x)}{16}$

Пример. Найти интеграл $\int \frac{tg(x)}{\left(1-ctg^2(x)\right)}dx$

```
(%i55) integrate(tan(x)/(1-(cot(x))^2), x);
```

(%o55) $\frac{\log(2\sin(x)^2-1)}{4}-\frac{\log(\sin(x)^2-1)}{2}$

Заключение.

Из всего выше написанного можно сделать следующие выводы: в современный учебный процесс внедряются новые методы обучения, которые возрождают достижения экспериментальной педагогики прошедшего столетия, которые построены на принципе саморазвития, активности личности. К одному из важнейших методов относится внедрение информационных технологий в обучении обучающихся и использование их на таких предметах, как математика.

Как уже говорилось ранее, Maxima — свободная система компьютерной алгебры. Это система для работы с символьными и численными выражениями, включающая: дифференцирование, интегрирование, разложение в ряд, преобразование Лапласа, обыкновенные дифференциальные уравнения, системы линейных уравнений, многочлены, множества, списки, векторы, матрицы и тензоры. Maxima производит численные расчеты высокой точности, используя точные дроби, целые числа и числа с плавающей точкой произвольной точности. Система позволяет строить графики функций и статистических данных в двух и трех измерениях.

Исходный код Maxima может компилироваться на многих системах, включая Windows, Linux и MacOS X. В отличии от более маститых требовательных и весомых Matlab и MathCad, Максима не требовательна к ресурсам, а самое главное совершенно бесплатна.

Максима – программа консольная, но к ней есть различные графические интерфейсы. Пожалуй, самый распространенный из них WxMaxima. Максима написана на языке Common Lisp.

Применение системы компьютерной математики Maxima позволяет:

- ✓ избавить студентов от множества рутинных вычислений и использовать дополнительное время для обдумывания алгоритмов решения задачи;
- ✓ повысить мотивацию обучения;
- ✓ проводить уроки на высоком эстетическом и эмоциональном уровне;
- ✓ обеспечить высокую степень дифференциации обучения;
- ✓ повысить объем выполняемой на уроке работы;
- ✓ усовершенствовать контроль знаний;
- ✓ рационально организовать учебный процесс, повысить эффективность урока;
- ✓ формировать навыки исследовательской деятельности;
- ✓ учитывать индивидуальные способности обучающихся;
- ✓ формировать информационную компетентность;
- ✓ развивать творческие способности и навыки самостоятельной продуктивной деятельности;
- ✓ применять интерактивность обучения;
- ✓ повышать интерес к математике и развивать ситуацию успеха каждого обучающегося;
- ✓ повысить качество усвоения программного материала и роста успеваемости по предмету.