Иванов Олег Александрович

MAXIMA В ОБУЧЕНИИ МАТЕМАТИКЕ В ШКОЛЕ

Этой работой автор начинает серию статей, в которых будет представлено описание свободно распространяемого пакета символьных вычислений Maxima и примеры его использования в преподавании математики в школе. Кроме того, автор попытается сформулировать некоторые методические принципы использования пакетов «компьютерной алгебры» при обучении школьников математике. Основная мысль состоит в том, что, поскольку целью всякого обучения является развитие обучаемых, то оценивать пригодность (и пользу) любого используемого в этом процессе дополнительного средства надо с точки зрения того, насколько его применение способствует достижению основной цели обучения. Часто приходится слышать, что компьютерные средства могут освободить учащегося от необходимости проводить рутинные вычисления и, тем самым, дать ему возможность больше думать. Однако вычисления становятся «рутинными» только тогда, когда ученик уже много-много раз проводил их самостоятельно. Для тех, кто только учится, решение каждой задачи – это небольшое, но открытие, маленький, но самостоятельный шаг. Поэтому учить «нажимать правильные кнопки», для того чтобы научить решать типовые учебные задачи, в высшей степени вредно. Компьютер должен помогать нам *рассуждать*.

Автор стремился к тому, чтобы содержание статей предлагаемой читателям серии могло служить основой для факультативных занятий по математике (и информатике!) с учащимися 10–11 классов. Примерные названия статей данной серии.

Урок 1. Введение в предмет.

Урок 2. Компьютерные эксперименты с Maxima.

Урок 3. Элементы программирования в Maxima.

Урок 4. Алгебра с Maxima.

Урок 5. Математический анализ с Maxima.

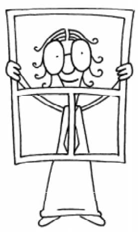
Урок 6. Заключение: обучение поиску решений задач с использованием компьютерных средств.

Итак, начнем.

УРОК 1. ВВЕДЕНИЕ В ПРЕДМЕТ

На этом уроке будут сформулированы первые синтаксические правила пакета Maxima, дано описание первых операторов этого пакета и приведены примеры их использования для решения двух не очень простых задач (находящихся, тем не менее, в рамках школьной программы).

Перед тем как читать основное содержание этой статьи, разумно установить на ваш компьютер одну из последних версий пакета Maxima, скачав дистрибутив с официального сайта www.sourceforge.net (автор пользуется версией wxMaxima, работающей под Windows). Тогда вы будете в состоянии повторять описанные в этой статье вычисления, при желании поэкспериментируете самостоятельно и, кроме того, сможете выполнить задания для самостоятельной работы.



Запустив программу wxMaxima, вы увидите стандартное окошко, в нижней строке которого через некоторое время справа появится фраза «Готова к вводу». Наберем с клавиатуры

```
>> sin(%pi/3);
```

и ничего не произойдет, поскольку мы пока не обратились к *ядру пакета*, который, собственно говоря, и производит вычисления. С этой целью нажмем клавишу ввода («Shift»+«Enter»), после чего на экране появятся:

```
(%i1) sin(%pi/3);
(%o1) sqrt(3)/2
```

строка ввода (*input*) за номером `(%i1)` и строка вывода (*output*) с номером `(%o1)`. В строке ввода находится исполняемый оператор, в строке вывода – результат его исполнения. В данном случае в строке вывода мы видим число $\frac{\sqrt{3}}{2} = \sin\frac{\pi}{3}$.

Прежде всего обратите внимание на то, что результатом является точное число, Maxima знает точное значение синуса $\frac{\pi}{3}$. С другой стороны, если мы обратимся к ней, попросив вычислить $\sin\frac{\pi}{5}$

```
(%i2) sin(%pi/5);
(%o2) sin(%pi/5)
```

,

то Maxima ничего не вычислит, поскольку она «не знает» этого значения – в отличие от пакета Mathematica, который даст ответ $\sqrt{\frac{5}{8} - \frac{\sqrt{5}}{8}}$. Конечно, Maxima знает его приближенное значение (см. рис. 1).

Заметим, что каждый из двух операторов в строке ввода `(%i3)` порождает свою строку вывода.

Сформулируем теперь первые *правила синтаксиса* пакета Maxima:

– аргумент оператора пишется в круглых скобках сразу после имени этого оператора;

– имена операторов, вычисляющих значения основных элементарных функций, похожи по написанию на стандартные математические обозначения соответствующих функций;

– различные операторы в строке ввода отделяются друг от друга *точкой с запятой*, при этом, если мы не хотим видеть на экране результат исполнения некоторого оператора, то после него следует поставить не точку с запятой, а *знак доллара*;

– со знака процента начинаются разнообразные служебные слова, в частности, имена стандартных математических констант, к примеру, `%pi` – это π, а `%e` – это *e*;

– в отличие от точки с запятой, играющей роль синтаксического разделителя, запятая ставится там, где идет перечисление (об этом – чуть дальше) или же делается уточнение; к примеру, `numer` в стро-

```
(%i3) sin(%pi/3),numer; sin(%pi/5),numer;
(%o3) 0.86602540378444
 (%o4) 0.58778525229247
```

Рис. 1

```
(%i5) tabsin:makelist(sin(k*%pi/6),k,0,6);
```

(%o5) $\left[0, \frac{1}{2}, \frac{\sqrt{3}}{2}, 1, \frac{\sqrt{3}}{2}, \frac{1}{2}, 0\right]$

```
(%i6) tabsin[2];
```

(%o6) $\frac{1}{2}$

Рис. 2

ке ввода `(%i3)` через запятую после имени каждого из операторов означает включение определенной опции, в данном случае – вычисления приближенного значения этого оператора.

Продолжим нашу первую сессию (см. рис. 2):

– знак *двоеточия* означает присвоение переменной с именем `tabsin` значения следующего за ней оператора;

– значением оператора `makelist` является некоторый *список*, в данном случае – список значений синусов углов вида $\frac{k\pi}{6}$ при $k = 0, 1, ..., 6$;

– *различные аргументы оператора* отделяются друг от друга запятыми, аргументами оператора `makelist` являются: выражение, имя итератора, наименьшее значение итератора, его наибольшее значение (по умолчанию итератор меняется от своего наименьшего значения до наибольшего с шагом 1);

– в квадратных скобках через запятую стоят элементы списка;

– для того чтобы взять некоторый элемент некоторого списка, надо после имени этого списка в квадратных скобках поставить его номер;

– *звездочка* означает знак умножения; в отличие от обычной записи математических выражений, при записи произведения в пакете Maxima опускать ее нельзя, `ab` – это переменная с именем *ab*, тогда как `a*b` есть произведение переменных с именами *a* и *b*.

Написав

```
(%i7) tabsin^2;
```

(%o7) $\left[0, \frac{1}{4}, \frac{3}{4}, 1, \frac{3}{4}, \frac{1}{4}, 0\right]$,

мы получили список, состоящий из квадратов элементов списка `tabsin`. Таким образом,

– обозначение `a^k`, как обычно, означает a^k – результат возведения числа в степень;

– при этом, если переменная *a* есть некоторый список, то результатом предыдущей операции является список, состоящий из степеней элементов исходного списка.

Операторы-функции, вычисляющие значения конкретных функций, составляют

```
(%i8) expand((1+sqrt(2))^2);factor(x^4+x^2+1);
      rat(1/(x*(x-1))+1/((x-1)*(x-2)));
```

(%o8) $2\sqrt{2}+3$

(%o9) $\left(x^2-x+1\right)\left(x^2+x+1\right)$

(%o10) $\frac{2}{x^2-2x}$

Рис. 3

```
(%i11) trigrat(4*cos(%pi/9)^3-3*cos(%pi/9));
(%o11) 1/2
```

Рис. 4

малую долю среди всех операторов в пакетах компьютерной алгебры. Основными являются операторы, *осуществляющие преобразования*. Преобразования алгебраических выражений осуществляют операторы, среди которых простейшими являются: `expand` – раскрытие скобок, `factor` – разложение на множители, `rat` – упрощение. На рис. 3 – примеры их использования.

Как вы видите, Maxima:

а) возвела в квадрат выражение $1+\sqrt{2}$, учтя при этом, что $\left(\sqrt{2}\right)^2=2$;

б) сумела разложить на множители многочлен степени 4;

в) сложила две алгебраические дроби, произведя при этом естественное сокращение. Для упрощения тригонометрических выражений есть специальный оператор `trigrat` (см. рис. 4).

Таким образом, хотя Maxima и не знает значения косинуса от $\pi/9$, однако она смогла свернуть данное ей выражение к значению $\cos\pi/3=1/2$.

Теперь давайте закончим нашу первую сессию, заставив Maxima «забыть все», после чего она, к примеру, не будет помнить значение списка `tabsin`.

```
(%i12) kill(all);
(%o0) done
```

Аргументами оператора `kill` являются имена, значения которых будут удалены из памяти, а если в качестве его аргумента взято `all`, то Maxima забывает значения всех имен.

Кроме того, теперь нумерация строк ввода и вывода снова начнется с 1.

Оператор `kill` стоит использовать с тем, чтобы в проводимых вычислениях при решении новой задачи не использовались значения ранее найденных переменных. Maxima, как всякий достаточно развитый пакет символьных вычислений, вполне может решать стандартные задачи как из

```
(%i1) solve([x+2*y=1,x^2+y^2=3],[x,y]);
```

$$(\%o1)\quad [\,[\,x=-\frac{2\sqrt{14}-1}{5},\ y=\frac{\sqrt{2}\sqrt{7}+2}{5}\,],\ [\,x=\frac{2\sqrt{14}+1}{5},\ y=-\frac{\sqrt{2}\sqrt{7}-2}{5}\,]\,]$$

```
(%i2) limit(sin(5*x)/log(2*x+1),x,0);
```

$$(\%o2)\quad \frac{5}{2}$$

```
(%i3) integrate(sin(x)^5,x,0,%pi/2);
```

$$(\%o3)\quad \frac{8}{15}$$

```
(%i4) rat(diff(sqrt(2*x+1)*x^3,x)),radcan;
```

$$(\%o4)\quad \frac{7x^3+3x^2}{\sqrt{2x+1}}$$

Рис. 5

школьных, так и из вузовских учебников и задачников. Надеюсь, что смысл приводимых ниже операторов понятен из их имен (см. рис. 5).

Оператор `solve` решает алгебраические уравнения и системы, при этом он имеет два аргумента. Первый из них – это список уравнений системы, второй – список переменных, относительно которых эту систему надо решать. Конечно, раз речь идет о списках, то их элементы в соответствии с общим правилом синтаксиса, должны быть заключены в квадратные скобки. Заметим только, что строка вывода `(%o1)` показана не полностью.

Оператор `limit` вычисляет пределы функций, его аргументами являются: выражение, предел которого надо найти, переменная, по которой ищется предел, значение переменной, к которому она стремится.

Оператор `integrate(f(x),x,a,b)` вычисляет определенный интеграл $\int_a^b f(x)dx$.

Наконец, оператор `diff` находит производную функции. Кстати, вот вам небольшое упражнение. Попробуйте поэкспериментировать, для того чтобы понять, с какой же целью автор в строке ввода `(%i4)` применяет оператор упрощения `rat`, включая при этом его опцию `radcan`.

В связи с приведенным примером возникает очень важная проблема, связанная с концептуальным вопросом – «Что есть образование, в нашем случае – математическое?». Нужно ли уделять в школе столько внимания преобразованиям выражений и правилам, по которым они проводятся, методам решения уравнений и их систем, правилам дифференцирования и так далее и тому подобное, если все это с легкостью делает компьютер? Можно ли разрешать школьникам и студентам пользоваться пакетами символьных вычислений? Что останется «в голове», если школьник для решения математической задачи будет садиться за клавиатуру?

В этой серии статей автор попытается показать, каковы должны быть методические идеи использования мощных компьютерных средств, чтобы они *помогали обучению математике*.

Обсуждение этой проблемы начнем с рассмотрения двух задач, при решении которых автор действительно использовал компьютер.

Задача 1. *Решите систему*

$$x=\frac{y}{1-y^2},\ y=\frac{z}{1-z^2},\ z=\frac{x}{1-x^2}.$$

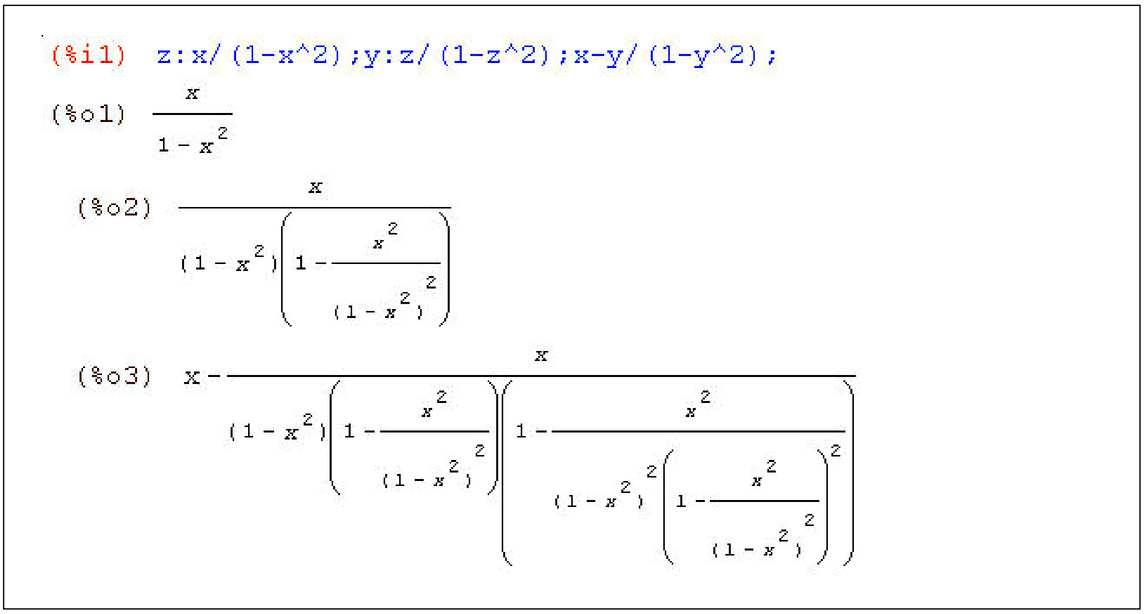


Рис. 6

Будем решать данную систему «методом подстановки»: выражение для y из второго уравнения подставим в первое, а в то, что в результате получится, подставим выражение для z из третьего уравнения данной системы. Конечно, полученное в результате этих подстановок уравнение надо еще и упростить. Все преобразования можно провести «на бумаге», но для увеличения скорости решения и гарантии от возможных ошибок обратимся к пакету Maxima. Сделаем подстановку (рис. 6) и упрощение

```
(%i4) rat(%);
```

$$(\%o4)\quad \frac{x^9-6x^7+9x^5-3x^3}{x^8-7x^6+13x^4-7x^2+1}$$

(конечно, разумнее было бы в строке `(%i1)` написать сразу `rat(x-y/(1-y^2))`.

Мы получили уравнение

$$x^9-6x^7+9x^5-3x^3=x^3\left(x^6-6x^4+9x^2-3\right)=0,$$

откуда $x=0$ или $x^6-6x^4+9x^2-3=0$. Решение $(x,y,z)=(0,0,0)$ было очевидно с самого начала, остается решить уравнение шестой степени, которое заменой $u=x^2$ сводится к кубическому уравнению $u^3-6u^2+9u-3=0$. Перед тем, как двигаться дальше, полезно посмотреть на график его левой части, с тем чтобы понять, сколько же корней оно имеет (рис. 7).

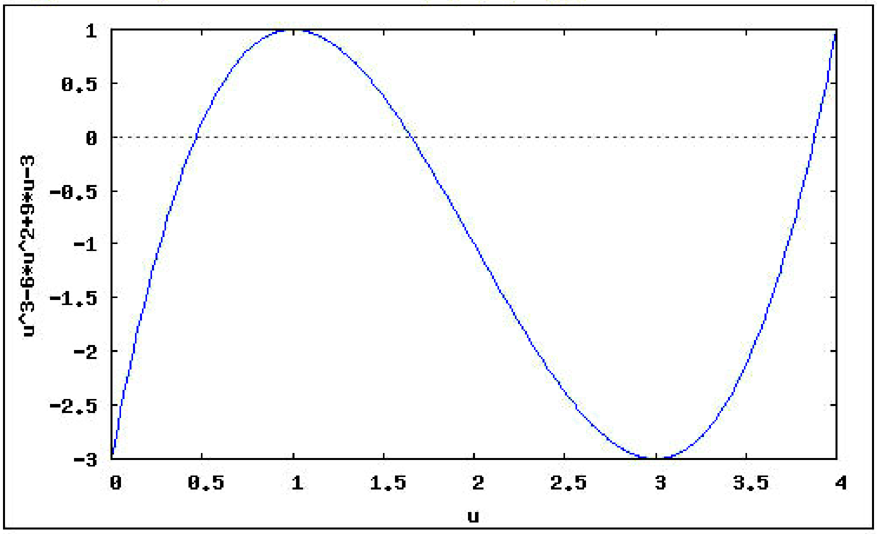
Графики функций одной переменной можно строить также при помощи оператора `plot2d`; отличие состоит в том, что в этом случае график будет находиться в отдельном окне.

Ясно видно, что корней три (обратите внимание, что ось x не совпадает с нижней границей рисунка) и все они положительны. Однако полученное уравнение не имеет рациональных корней, так что «подбирать» их бессмысленно, даже имея в своем распоряжении компьютер. Можно найти их приближенно, однако первая попытка воспользоваться опцией `numer` в операторе `solve` дает странный ответ – убедитесь в этом сами. Для вычисления приближенных значений корней уравнения (функции) применяется оператор `allroots` (рис. 8).

Поскольку нас интересуют значения $x=\pm\sqrt{u}$, найдем и их (см. рис. 9).

Знак процента в аргументе оператора sqrt извлечения квадратного корня означает, что этот оператор применяется к результату исполнения предыдущего оператора.

```
(%i5) wxplot2d(u^3-6*u^2+9*u-3,[u,0,4]);
(%t5)
```




```
(%o5)
```

Рис. 7

```
(%i7) allroots(u^3-6*u^2+9*u-3);
(%o7) [ u = 0.46791111376204 , u = 1.65270364466614 , u = 3.879385241571816 ]
```

Рис. 8

```
(%i8) sqrt(%);
(%o8) [ sqrt(u) = 0.68404028665134 , sqrt(u) = 1.285575219373079 , sqrt(u) = 1.969615506024416 ]
```

Рис. 9

Можно было дать явную ссылку на строку вывода, указав в качестве его аргумента `%o7`.

Кстати, надеюсь, что вас не удивляет, что мы применили оператор `sqrt` *к списку*?

Продолжим решение. Следующим (естественным) шагом является сведение полученного кубического уравнения к уравнению вида $x^3+px+q=0$ – так называемому приведенному кубическому уравнению. Поскольку

$$u^3-6u^2+9u-3=(u-2)^3-12u+8+9u-3=$$

$$=(u-2)^3-3(u-2)-1,$$

то для этого следует сделать замену $w=u-2$. Чтобы проверить свои вычисления, обратимся к Maxima (см. рис. 10).

Все верно, мы пришли к уравнению $w^3-3w-1=0$. Не кажется ли вам, что вы уже видели похожее выражение?! Из результата исполнения приведенного выше оператора строки `(%i11)` следует, что одним из корней уравнения $4x^3-3x=\frac{1}{2}$ является число $\cos\pi/9$. В нашем случае коэффициент при w^3 равен 1, а не 4. Однако замена w = 2v сведет наше уравнение к известному! Положив w = 2cos$t$, получив уравнение $8\cos^3 t-6\cos t-1=0$, или $4\cos^3 t-3\cos t=\frac{1}{2}$, или $\cos 3t=\frac{1}{2}$, откуда $3t=\pm\frac{\pi}{3}+2\pi k$, или $t=\pm\frac{\pi}{9}+\frac{2\pi k}{3}$, где $k\in Z$.

Следовательно, корнями уравнения являются числа $w_k=2\cos\left(\frac{\pi}{9}+\frac{2\pi k}{3}\right)$, k = 0, 1, 2, а корнями исходного уравнения шестой степени – числа

$$x=\pm 2\cos\frac{\pi}{18},\pm 2\cos\frac{7\pi}{18},\pm 2\cos\frac{13\pi}{18}.$$

Ответ можно записать в другой форме, воспользовавшись формулами приведения. Корнями исходного уравнения степени 9 являются числа x = 0 и $x=2\cos\frac{(6k+1)\pi}{18}$, где k = 0, 1, 2, ..., 5. Однако исходная задача не решена, поскольку еще следует вычислить значения $z_k=\frac{x_k}{1-x_k^2}$ и $y_k=\frac{z_k}{1-z_k^2}$, что не совсем тривиально. Ясно, что соответствующие значения переменных y и z должны совпасть с одним из значений переменной x, но с каким именно?! «Озадачьте» погрумму Maxima этим вопросом (ответ будет приведен в следующей статье).

Может показаться, что нам просто повезло найти удачную замену в уравнении относительно w, однако справедлив следующий факт. Если кубическое уравнение имеет три действительных корня, то они могут быть явно выражены через три-

```
(%i9) u:w+2$ expand(u^3-6*u^2+9*u-3);
(%o10) w^3 - 3 w - 1
```

Рис. 10

гонометрические (и обратные тригонометрические) функции, см., к примеру, [1: с. 112–113].

Итак, в решении задачи 1 пакет Maxima избавил нас от рутинных вычислений. Следующая задача также является чисто математической. В отличие от предыдущего рассуждения, решение, которое здесь будет получено, никак не будет опираться на «компьютерный счет». Его роль, однако, очень важна, и, надеюсь, будет понятна из последующего обсуждения.

Задача 2. *Вычислите сумму*

$$ctg^2\frac{\pi}{2n+1}+ctg^2\frac{2\pi}{2n+1}+\ldots+ctg^2\frac{n\pi}{2n+1}.$$

Вначале немного посчитаем. При $n = 1$ имеется единственное слагаемое $ctg^2\frac{\pi}{3}=\frac{1}{3}$. При $n = 2$ можно использовать явные формулы (но Maxima их «не знает»), $ctg\frac{\pi}{5}=\sqrt{1+\frac{2}{\sqrt{5}}}$ и $ctg\frac{2\pi}{5}=\sqrt{1-\frac{2}{\sqrt{5}}}$, поэтому $ctg^2\frac{\pi}{5}+ctg^2\frac{2\pi}{5}=2$. Обратимся к пакету (см. рис. 11).

```
(%i1) cot(%pi/5)^2+cot(2*%pi/5)^2;
```

$$\text{(\%o1)}\quad \cot\left(\frac{2\,\%pi}{5}\right)^2+\cot\left(\frac{\%pi}{5}\right)^2$$

Рис. 11

Конечно, cot – это оператор, вычисляющий значения котангенса.

Приближенное значение этой суммы Maxima в состоянии найти.

```
(%i2) %,numer;
(%o2) 2.0
```

Составим список из приближенных значений искомых сумм для n = 2, 3, 4, 5 (см. рис. 12).

Оператор sum(*expr,i,imin,imax*) вычисляет сумму, слагаемыми которой являются значения выражения *expr*, при значениях итератора *i*, меняющихся от *imin* до *imax* (с шагом 1).

Таким образом, очень похоже на то, что точными значениями при n = 3, 4, 5 являются, соответственно, 5, $\frac{28}{3}$ и 15. А про какой же оператор пакета Maxima «забыл» автор? Конечно, можно было сразу использовать оператор trigrat (см. рис. 13).

Итак, ответ в частных случаях получен, хотя в данный момент мы не имеем ни малейшего представления ни о том, каким же это было сделано образом, ни – и это главное – какова же общая формула. Попробуем догадаться. Как вы видите, при $n = 1$ и $n = 4$ ответ – рациональное число со знаменателем 3.

Записав каждую из найденных сумм в виде дроби с тройкой в знаменателе, мы получим дроби

```
(%i3) makelist(sum(cot(k*%pi/(2*n+1))^2,k,1,n),n,2,5),
      numer;
(%o3) [2.0,5.000000000000001,9.333333333333336,15.0]
```

Рис. 12

```
(%i4) makelist(trigrat(sum(cot(k*%pi/(2*n+1))^2,k,1,n)),
      n,2,5);
```

$$\text{(\%o4)}\quad [2,5,\frac{28}{3},15]$$

Рис. 13

$\frac{1}{3}=\frac{1\cdot 1}{3}$, $2=\frac{2\cdot 3}{3}$, $5=\frac{3\cdot 5}{3}$, $\frac{28}{3}=\frac{4\cdot 7}{3}$, $15=\frac{5\cdot 9}{3}$

Не очевидно ли, что в числителях этих дробей стоят произведения $n\cdot(2n-1)$? Естественно предположить, что имеет место формула

$$ctg^2\frac{\pi}{2n+1}+ctg^2\frac{2\pi}{2n+1}+\ldots+ctg^2\frac{n\pi}{2n+1}=\frac{n(2n-1)}{3}.$$

Обратимся к пакету Maxima (см. рис. 14) при n = 6, ожидая ответ: 22.

Получилось! Теперь не остается никаких сомнений в справедливости найденной формулы, хотя ее еще предстоит доказать. Идея доказательства никак не связана с преобразованиями тригонометрических выражений, а имеет более общий характер. Мы уже видели, что значения тригонометрических функций удовлетворяют определенным алгебраическим соотношениям. Это и есть ключ к решению – написать алгебраическое уравнение, корнями которого являются участвующие в сумме котангенсы.

Всем известно, что $\sin 2x=2\cos x\sin x$ и $\sin 3x=3\cos^2 x\sin x-\sin^3 x$. Аналогичную формулу можно написать и в общем случае. Оказывается, что (см., к примеру, [1: с. 322–324])

$$\sin(2n+1)x=(2n+1)\cos^{2n}x\sin x-$$
$$-\frac{n(2n-1)(2n+1)}{3}\cos^{2n-1}x\sin^3 x+\ldots$$
$$+(-1)^n\sin^{2n+1}x.$$

Так как уравнение $\sin(2n+1)x=0$ имеет своими корнями числа $x_k=\frac{k\pi}{2n+1}$, $k\in Z$, то эти же числа являются корнями уравнения

$$(2n+1)\cos^{2n}x\sin x-\frac{n(2n-1)(2n+1)}{3}\cos^{2n-1}x\sin^3 x+\ldots+(-1)^n\sin^{2n+1}x=0.$$

Поделив его левую часть на $(2n+1)\sin^{2n+1}x$, мы получим уравнение

$$ctg^{2n}x-\frac{n(2n-1)}{3}ctg^{2n-2}x+\ldots+\frac{(-1)^n}{2n+1}=0$$

с корнями $x_k=\frac{k\pi}{2n+1}$, где k – это целое число, не кратное $2n$ + 1. Замена $t=ctg^2 x$ приводит к алгебраическому уравнению

$$t^n-\frac{n(2n-1)}{3}t^{n-1}+\ldots+\frac{(-1)^n}{2n+1}=0.$$

Как следует из предыдущего рассуждения, его корнями являются числа $t_k=ctg^2\frac{k\pi}{2n+1}$, где k = 1, 2, ..., n. В силу обобщенных формул Виета (см., к примеру, [1: с. 127]), сумма корней противоположна коэффициенту при t^{n-1}, так что $t_1+t_2+\ldots+t_n=\frac{n(2n-1)}{3}$. Искомая формула доказана.

Подведем итог.

Смысл вычислений, проведенных при помощи пакета Maxima, состоял в том, что этот своего рода «компьютерный эксперимент» указал нам явно на итоговую формулу. А если вы уверены в справедливости некоторой формулы (утверждения), то идея доказательства появляется в вашем мозгу намного быстрее, чем если бы вы в ней (в нем) сомневались.

Заключительное замечание.

Конечно, мы могли, не вычисляя приближенных значений сумм (см. `(%i3)`), узнать их точные значения (написав стро-

```
(%i5) n:6$ trigrat(sum(cot(k*%pi/(2*n+1))^2,k,1,n));
(%o6) 22
```

Рис. 14

ку (%i4)). Однако дело в том, что если мы можем быть уверены, что приближенные значения Maxima найдет быстро, то изначально мы не знаем, сможет ли она найти их точно – это гораздо более сложная задача. Разница еще и во времени исполнения. Результатом исполнения оператора time является время, затраченное процессором на вычисление соответствующего значения. Как вы видите, разница – разительная.

```
(%i7)  time(%o3,%o4);
(%o7)  [0.01, 0.46]
```

Задания для самостоятельной работы

1. Вычислите произведение

$$tg\frac{\pi}{2n+1}\cdot tg\frac{2\pi}{2n+1}\cdot\ldots\cdot tg\frac{n\pi}{2n+1}.$$

2. Найдите решения системы задачи 1.

3. Решите уравнение $x^4+4x-1=0$ (записав решение на бумаге).

4. Числами Фибоначчи F_n (n = 1, 2, ...) называются числа последовательности, в которой F_1 = 1 и F_2 = 1, а все последующие числа строятся по правилу $F_n = F_{n-1} + F_{n-2}$. Значением оператора fib(n) пакета Maxima является n-ое число Фибоначчи. Проведите «компьютерный эксперимент», в результате которого догадайтесь, какой будет сумма n первых чисел Фибоначчи. Докажите найденную формулу.

5. Добавление опции simp (упрощение) к оператору sum (то есть использование оператора simpsum) в некоторых случаях обеспечивает нахождение явных формул для сумм. Найдите и докажите формулу для сумм

а) квадратов;

б) кубов n первых натуральных чисел.

Литература

1. *О.А. Иванов.* Задачи по алгебре и началам анализа. СПб: БХВ-Петербург, 2005. 380 с.

Иванов Олег Александрович, ***профессор, доктор педагогических наук и кандидат физико-математических наук, профессор кафедры общей математики и информатики СПбГУ.***