

Обмен опытом и передовые идеи по использованию свободного ПО в образовании

## Компьютерная алгебра

Машины пока не умеют думать – это прерогатива человека. Но они хорошо справляются с рутинной работой, например, дифференцированием и решением уравнений, утверждает **Семен Есилевский**.



Наш эксперт

**Семен Есилевский**  
Научный сотрудник, вычислительный программист и энтузиаст Linux и Open Source. Кандидат физико-математических наук, работает в области теоретической биофизики и молекулярного моделирования.

На уроках математики в школе учителя любят повторять, что компьютер никогда не заменит человека в аналитических преобразованиях математических выражений. В какой-то степени это правда: только человек может сформулировать цель, которую необходимо достичь. Тем не менее большая часть аналитической работы в математике является рутинной – дифференцирование, интегрирование, разложение в ряды и так далее проводится по четким алгоритмам (которые, однако, могут быть очень сложными) или готовым таблицам. Именно эту рутинную работу и призваны облегчить системы компьютерной алгебры (Computer Algebra Systems – CAS). CAS по праву считаются одними из самых сложных в алгоритмическом плане программ. С их разработкой и поддержкой могут справиться только высококлассные специалисты. Не удивительно, что самые известные CAS – *Wolfram Mathematica*, *Maple*, *Mathcad* – проприетарны и дороги. Высокие требования к разработчикам делают создание CAS «с нуля» очень сложной задачей для энтузиастов свободного ПО. Большинство из существующих открытых CAS являются наследниками коммерческих систем, которые по тем или иным причинам потеряли рыночную ценность и были переведены в ранг проектов с открытым кодом. Это не означает, что они являются ущербными – причины ухода с рынка обычно носят чисто финансовый характер. Но, к сожалению, полноценных свободных аналогов коммерческих CAS на сегодняшний день не существует. Однако возможностей рассмотренных в этой статье систем должно быть достаточно для большинства практических задач.

«Только человек может сформулировать необходимую цель.»

Сравнивать открытые и коммерческие CAS можно академически (количество поддерживаемых математических абстракций, качество и строгость преобразований) или с точки зрения конечного пользователя (удобство интерфейса и синтаксиса, возможности построения графиков и экспорта результатов, уровень системы помощи). В этой статье мы будем смотреть на мир глазами человека, не являющегося профессиональным математиком (школьника старших классов, студента, инженера или научного работника нематематической специальности). Для таких людей, как правило, достаточно базовых аналитических преобразований, имеющихся в любой из рассмотренных CAS. Мы затронем интерфейс, стабильность работы, графические возможности, систему

помощи и наиболее заметные особенности синтаксиса команд.

Нужно подчеркнуть, что CAS – это специализированные системы символьной алгебры. Пакеты численной математики, такие как *Octave*, могут содержать модули для работы с символьными преобразованиями, но они не являются для них основными и в этой статье не рассматриваются.

### Maxima

Система *Maxima* ([maxima.sf.net](http://maxima.sf.net)) является, пожалуй, самой популярной открытой CAS. Она появилась в 1982 году как открытая ветвь старейшей из всех CAS – *Macsyma*, берущей начало в 1960-х. Проект активно развивается и поддерживается силами сообщества. Сама по себе *Maxima* – это полностью кроссплатформенное командно-строчное аналитическое ядро, написан-

Хотите завести свою Linux-библиотеку?  
Подробности на стр. 107



ное на Lisp. Работать с ним напрямую неудобно, и в подавляющем большинстве случаев используются различные оболочки, самая популярная из которых – *wxMaxima* ([wxmaxima.sf.net](http://wxmaxima.sf.net)). Она кросс-платформенная, базируется на *wxWidgets* и присутствует в репозиториях всех основных дистрибутивов Linux (а также доступна для Windows и Mac OS X).

*wxMaxima* реализует концепцию документов, состоящих из ячеек [cell]. Ячейки могут содержать текст, формулы или графики. Если просто начать набирать текст в свободном месте рабочего листа, создается математическая ячейка (другие типы доступны в меню Edit > Cell > New...). В каждой математической ячейке есть строка ввода и область вывода. Чтобы передать введенную строку ядру для обработки, нужно нажать Shift+Enter, после чего в области вывода появляется результат вычислений. Ячейки можно сворачивать, скрывая громоздкие результаты, щелкнув на треугольнике в верхней части «скобки». Их также можно копировать, удалять и перемещать произвольным образом. При редактировании строки ввода *wxMaxima* подсвечивает совпадающие скобки, дополняет ключевые слова и при необходимости сама добавляет завершающую точку с запятой.

Под рабочей областью находится ряд кнопок для наиболее часто выполняемых операций и преобразований. Многие из них открывают диалоги, позволяющие указывать параметры без необходимости помнить синтаксис конкретной команды. В меню содержатся практически все доступные в *Maxima* преобразования, также с возможностью указания параметров в диалогах.

Нужно отметить значительный прогресс в интуитивности и удобстве интерфейса *wxMaxima*, достигнутый в последнее время. В предыдущих версиях этой оболочки единая строка ввода располагалась внизу рабочей области, а редактирование внутри самих ячеек было очень неудобным. В новых версиях наконец-то свершился переход к единообразному редактированию непосредственно внутри ячеек. Это сближает интерфейс *wxMaxima* и коммерческих CAS, особенно *Mathematica* и *Maple*.

Система помощи организована достаточно стандартно. Кроме подробной документации по *Maxima*, можно запросить пример использования конкретной команды и список команд, включающих ключевое слово. В то же время встроенной справки по работе с самой *wxMaxima* нет, хотя на официальном сайте есть множество учебных материалов, устраняющих этот недостаток.

Мы детально рассматривали *Maxima* в статьях **LXF81–86** ([wiki.linuxformat.ru](http://wiki.linuxformat.ru)), поэтому сейчас начнем с простого примера разложения многочлена на множители (линии – границы ячеек):

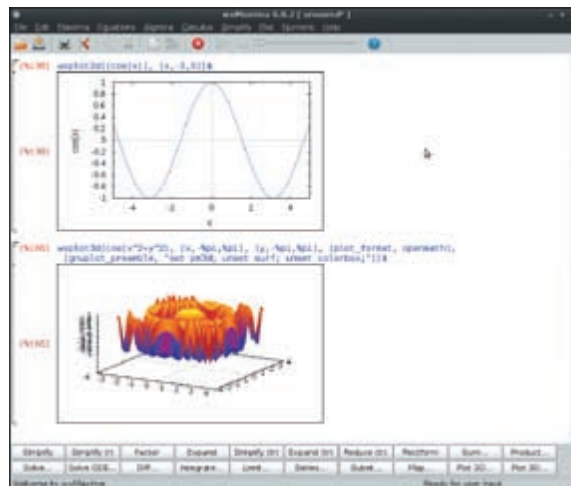
```
(%i1) (x+y)^6;
(%o1) (y+x)^6
(%i2) expand(%);
(%o2) y^6+6*x*y^5+15*x^2*y^4+20*x^3*y^3+15*x^4*y^2+6*x^5*y+x^6
(%i3) factor(%);
(%o3) (y+x)^6
```

Результат предыдущего вычисления сохраняется в универсальной для многих CAS «магической» переменной %. Интегрирование и дифференцирование тоже выполняются тривиально:

```
(%i10) f: cos(x^2);
(%o10) cos(x^2)
(%i12) g: diff(f,x);
(%o12) -2*x*sin(x^2)
(%i13) integrate(g, x);
(%o13) cos(x^2)
```

В этом примере показано применение символических имен (f и g), которые можно затем использовать в других выражениях.

*wxMaxima* может отображать двух- и трехмерные графики. Для этого привлекается внешняя программа (*Gnuplot/OpenMath*). Стандартные функции рисования графиков *Maxima* (plot2d и plot3d) в *wxMaxima* заменяются на специфические для этой оболочки wxplot2d и wxplot3d. Примеры графиков, построенных с помощью



► Рис. 1. Интерфейс *wxMaxima*. Показаны примеры двумерного и трехмерного графиков.

*Gnuplot*, представлены на рис. 1. Они могут вставляться в виде графического файла непосредственно в документ или отображаться в отдельном окне. О возможностях интерактивного редактирования графиков, которое является одной из самых ценных функций коммерческих CAS, в *wxMaxima*, естественно, нет и речи. Все параметры нужно задавать непосредственно в аргументах функций wxplot2d и wxplot3d. К сожалению, назвать их интуитивно понятными сложно. К тому же разные аспекты внешнего вида графика совершенно неочевидным образом разбросаны между аргументами wxplot2d/3d и дополнительными командами, передаваемыми внешней программе. Так, в случае *Gnuplot*, чтобы получить график приемлемого качества, нужно хорошо знать многочисленные внутренние команды (**LXF122**). Зато, при наличии опыта, можно получать графики полиграфического качества.

Поддержка кириллицы в *wxMaxima* реализована прекрасно. Русские буквы корректно отображаются не только в текстовых ячейках и комментариях, но и на графиках *Gnuplot*. Последнее действительно является достижением, так как *Gnuplot* в интерактивном режиме славится своей несовместимостью с кириллицей.

Стабильность работы связки *Maxima+wxMaxima* в основном не вызывает нареканий. Однако нам приходилось неоднократно наблюдать фатальные ошибки интерпретатора при интегрировании и построении графиков. К счастью, они обычно устраняются после перезапуска интерпретатора (Maxima > Restart Maxima).

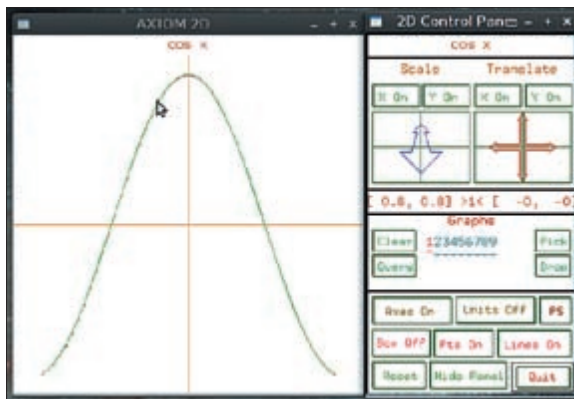
## Axiom

*Axiom* ([www.axiom-developer.org](http://www.axiom-developer.org)) – свободная CAS общего назначения, появившаяся в далеком 1971 году в исследовательских лабораториях IBM. В 2001 году программа стала распространяться свободно по лицензии типа BSD. *Axiom* претендует на звание универсальной системы математического программирования, в которой естественные математические понятия, такие как полиномы и дроби, являются строго определенными типами данных. *Axiom* работает в любых Unix-подобных системах и присутствует в репозиториях всех основных дистрибутивов Linux, но его жесткая зависимость от X-сервера затрудняет перенос на другие системы, в частности, Windows. *Axiom* можно запустить и там, но придется использовать специальный X-сервер для Windows.

*Axiom* долгое время находилась в состоянии стагнации. Система была стабильна и функциональна, но интерфейс все больше и больше морально устаревал, а модернизация и расширение функциональности сдерживались как техническими причинами, так и сообществом разработчиков. В результате в 2007 году от *Axiom* откололись два самостоятельных проекта *OpenAxiom* ([www.open-axiom.org](http://www.open-axiom.org)) и *Fricas* ([fricas.sourceforge.net](http://fricas.sourceforge.net)), целью ко-

»

► Рис. 2. Двумерный график с контрольной панелью в *Axiom*.



которых является дальнейшее развитие и модернизация системы. Наличие трех параллельных проектов создает определенную путаницу, однако с точки зрения пользователя отличия между ними на сегодняшний день практически отсутствуют.

Графического интерфейса как такового у *Axiom* нет. Имеется командно-строчный интерпретатор, система вывода документации HyperDoc и система отображения графиков. Две последние базируются непосредственно на X-сервере и выглядят, мягко говоря, архаично и непривлекательно. Кроме этого, *Axiom* можно использовать в научном редакторе *TeXmacs*.

Базовый синтаксис *Axiom* очень схож с *Maxima*:

```
(3) -> expand((x+y)^6)
(3) ->
      6      5      2 4      3 3      4 2      5      6
(3) y + 6x y + 15x y + 20x y + 15x y + 6x y + x
Type: Polynomial Integer

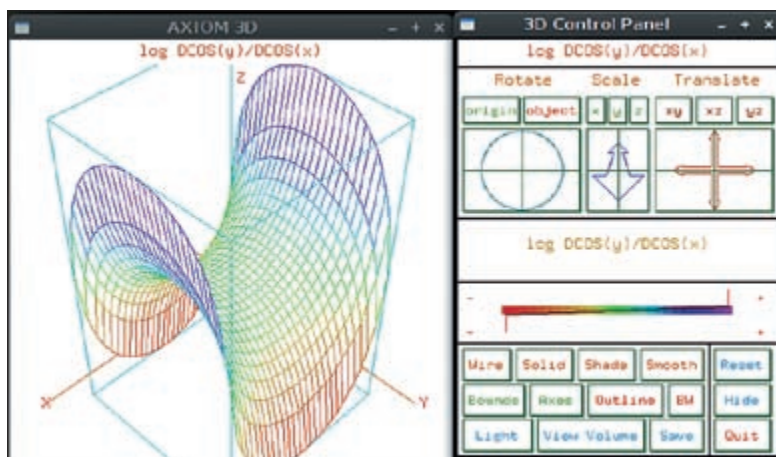
(4) -> factor(%)
(4) ->
      6
(4) (y + x)
Type: Factored Polynomial Integer

Используется уже знакомая «магическая» переменная %, содержащая последний вычисленный результат, и та же последовательная нумерация строк ввода и вывода. Интегрирование и дифференцирование также похожи:

(3) -> D(cos(x),x)
(3) ->
(3) - sin(x)
Type: Expression Integer

(4) -> integrate(%,x)
(4) ->
(4) cos(x)
Type: Union(Expression Integer,...)
```

► Рис. 3. Трехмерный график с контрольной панелью в *Axiom*.



Переменные вводятся практически идентично:

```
(5) -> a:=cos(x^2)
(5) ->
      2
(5) cos(x)
Type: Expression Integer

(6) -> D(a,x)
(6) ->
      2
(6) - 2x sin(x)
Type: Expression Integer
```

Желая организовать в *Axiom* подобие работы с целостными документами, имеет смысл сохранить последовательность выражений в текстовый файл и интерпретировать его целиком.

За построение двумерных и трехмерных графиков в *Axiom* ответственна единая команда `draw`. Если передается функция одной переменной, то строится двумерный график:

```
(3) -> draw(cos(x),x=-3..3)
```

График выводится в отдельном окне. При щелчке по нему открывается контрольная панель, позволяющая менять масштаб и положение отображаемой области графика, выводить или прятать оси, подписи и так далее (рис. 2)

Если строится график функции двух переменных, например,

```
(4)-> draw(log(cos(y)/cos(x)),x=-1.5..1.5,y=-1.5..1.5)
```

открывается соответствующая контрольная панель трехмерного графика с дополнительными опциями отображения заливки и скрытых поверхностей, а также с возможностью свободного вращения (рис. 3).

В целом качество графики в *Axiom* невысокое: нет сглаживания линий и шрифтов, отсутствует плавная градиентная заливка поверхностей. Контрольные панели выглядят архаично и неинтуитивно. Масштабирование и свободное вращение реализованы неудачно – соответствующие инструменты излишне чувствительны.

При использовании *Axiom* в Ubuntu (и, видимо, во всех других производных Debian) необходимо запускать интерпретатор с явным указанием архитектуры: `setarch i386 -R axiom`, иначе он не находит необходимых библиотек.

В целом *Axiom* демонстрирует, как выглядели CAS два десятилетия назад. При наличии полной математической функциональности архаичный интерфейс делает эту программу непривлекательной для интерактивного применения. Однако крайне высокие системные требования позволяют использовать эту систему даже на самых слабых компьютерах. *Axiom* также успешно применяется во «встроенном» виде как мощное аналитическое ядро, в *TeXmacs* и в среде *Sage* ([www.sagemath.org](http://www.sagemath.org)).

## Yacas

*Yacas* ([yacas.sf.net](http://yacas.sf.net)), как следует из ее названия, это «еще одна система компьютерной алгебры» (Yet Another Computer Algebra System). Она использует собственный язык программирования, разработанный для символьных вычислений и операций с числами произвольной точности. По сравнению с другими CAS, которые нередко имеют более чем двадцатилетнюю историю, *Yacas* – совсем молодая система, развиваемая небольшой группой энтузиастов. Естественно, она пока не может считаться зрелой, и в ней отсутствуют многие возможности других CAS. Тем не менее, для рутинных аналитических преобразований возможностей *Yacas* вполне хватает. *Yacas* может работать локально, как командно-строчный интерпретатор, однако удобнее использовать ее как интернет-сервис. На сайте проекта в систему можно войти сразу же, без предварительной регистрации. На вкладке Tutorial имеется учебник, описывающий синтаксис, основные возможности и принципы языка программирования *Yacas*. Вкладка My Yacas представляет собой привычный интерактивный интерпретатор, реализованный в виде Java-апплета (рис. 4). При вводе

команд срабатывает очень полезная и удобная функция автодополнения, которая не только экономит время, но и подсказывает, какие аргументы принимает та или иная функция. В правой части страницы находится список всех доступных функций, для каждой из которых отображается краткая справка с примерами, которые можно отправить в интерпретатор одним щелчком мыши. Большим недостатком является невозможность копирования текста уже выполненных команд и, что самое неудобное, их результатов. В какой-то степени это компенсируется наличием истории в строке ввода. Неприятной особенностью является также невозможность работать с кириллицей (Java-апплет просто не воспринимает русские буквы). В целом web-интерфейс Yacas производит двоякое впечатление. Он не предоставляет функциональности «документа», а является всего лишь интерактивным интерпретатором. В то же время, он достаточно отзывчив, не требует регистрации и прекрасно подходит для выполнения простых операций «на лету».

Синтаксис Yacas в некоторых аспектах довольно непривычен. Имена всех функций начинаются с заглавной буквы (например, Sin), но в строках вывода с результатами вычислений они пишутся в обычной нотации (sin). Выражение, к которому применяется некая функция, в ряде случаев передается как аргумент, а иногда просто пишется после функции. Например, дифференцирование и интегрирование выглядят следующим образом:

```
In> a:=Cos(x)
```

```
In> D(x) a
```

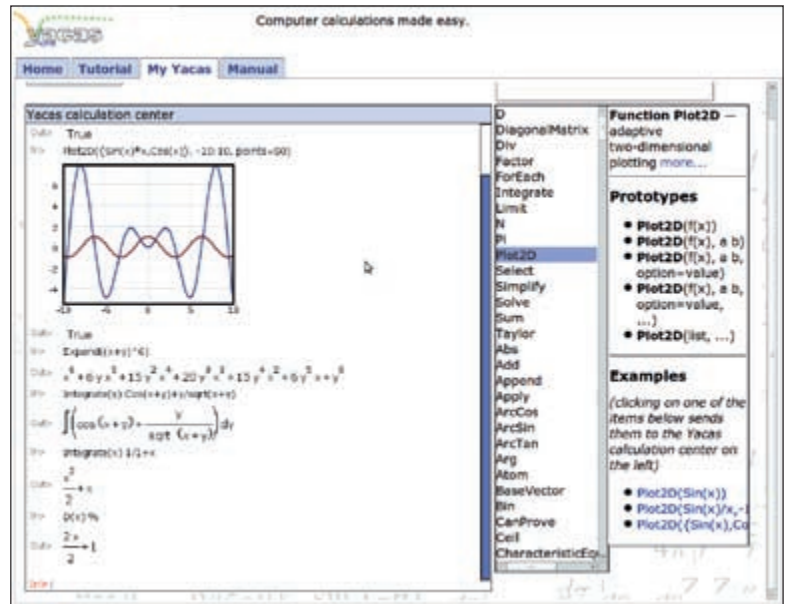
```
Out> -sin(x)
```

```
In> Integrate(x) a
```

```
Out> sin(x)
```

Впрочем, наличие отличной системы помощи во многом сглаживает эти шероховатости.

Графические возможности Yacas довольно скромны. Имеется функция Plot2D для построения обычных двумерных графиков. Они выглядят достаточно привлекательно, но количество доступных опций форматирования очень невелико. Возможности построения параметрических, контурных и других графиков отсутствуют. Для визуализации трехмерных поверхностей предусмотрена функция Plot3DS, однако она недоработана и способна только выводить числовые данные для графика в виде текстового файла, который затем можно обработать сторонней программой.



В целом проект Yacas еще «сырой», но имеет прекрасный потенциал для развития и уже сейчас может применяться как простой и удобный «аналитический калькулятор».

## Итоги

Несмотря на объективные сложности разработки систем символической алгебры силами энтузиастов, развитие открытых CAS не стоит на месте. Однозначного лидера среди них назвать трудно: выбор сильно зависит от предполагаемой модели использования. Если необходима система, максимально приближенная к коммерческим CAS по методам работы, интерфейсу, синтаксису и аналитическим возможностям, оптимален тандем *Maxima+wxMaxima*. Система *Axiom* имеет отличную математическую функциональность, а ее требования настолько скромны, что ее можно применять даже на очень слабых машинах, которых немало в школах и вузах; правда, приходится мириться с архаичностью интерфейса. Для быстрого выполнения нескольких аналитических преобразований «на лету» и построения простых двумерных графиков прекрасно подходит интернет-сервис Yacas.

► Рис. 4. Web-интерфейс Yacas. Показан двумерный график и примеры некоторых преобразований.

## Что было: Фестиваль свободного ПО в Красноярске

Детский компьютерный фестиваль – добрая традиция в Красноярске. В этом году он прошел в семнадцатый раз и был посвящен теме «Свободное программное обеспечение». Организаторы – Главное управление образования администрации г. Красноярск и городской Координационный совет по информатизации образования – рассчитывали, что он станет серьезным подспорьем при переходе на СПО, как для учителя, так и для школы. О ходе и результатах Фестиваля рассказывает председатель Красноярского городского координационного совета по информатизации Евгений Васильевич Кузнецов.

Фестиваль проходил в несколько этапов. В конце сентября было выбрано и подготовлено 12 гранд-тьюторов по СПО. Обучение велось Академией Айти. Гранд-тьюторами, в свою очередь, были проведены курсы для тьюторов по прикладным направлениям СПО. Всего обучено прошло 105 человек.

В связи с эпидемией гриппа основная часть Фестиваля проходила дистанционно. Школьным командам случайной жеребьевкой были розданы настольные игры. Задание состояло в том, чтобы средствами СПО создать для полученной игры полную информационную среду (аннотацию, правила, презентацию, видеообу-

чалку, буклет, плакат и т.п.). Результаты работ принимались только в свободных форматах и размещались на [wiki.kkr.ru](http://wiki.kkr.ru). В дистанционном этапе приняло участие 1080 человек из 94 школ г. Красноярск, 5 школ Пермского края, 5 школ Томской области. Работы оценивались экспертным комитетом, в который вошли 15 экспертов по СПО (по пять из Красноярск, Томск, Пермь) и 15 экспертов по содержательной части.

В результате Фестиваля были награждены 43 школьные команды, из них одна – команда гимназии № 2 г. Красноярск – признана абсолютным победителем.

Фестиваль показал, что переход на СПО в Красноярске будет проходить сложно. Учителя и ученики привыкли к определенному интерфейсу, комфорту при работе с прикладными программами. С другой стороны, уровень работ лучших команд был достаточно высок. Различить на глаз, какими средствами делались лучшие видеоролики, буклеты или плакаты, вряд ли смогут даже специалисты. Это дарит надежду, что и у нас появятся в школах СПО-энтузиасты – учителя, фанаты – ученики.

В заключение хотелось бы поблагодарить партнеров Фестиваля – компании Альт-Линукс, ТТК, Академию Ай-

Ти, наших коллег из Томска, Перми и Красноярск за активную помощь в организации и проведении мероприятий, экспертизе детских работ, а, главное, за поддержку всех наших дел, которую мы ощущали на протяжении четырех месяцев. LXF



► Команда гимназии № 2 г. Красноярск принимает поздравления от Е.В. Кузнецова.