Записки научных
семинаров ПОМИ
Том 281, 2001 г.

В. И. Мысовских

СИСТЕМЫ КОМПЬЮТЕРНОЙ АЛГЕБРЫ И СИМВОЛЬНЫЕ ВЫЧИСЛЕНИЯ

§1. Введение

Компьютерная алгебра – это новое направление, возникшее при взаимодействии ряда математических дисциплин (в первую очередь, алгебры) и информатики. В широком смысле под этим словосочетанием понимаются любые символьные (в отличие от численных) вычисления, выполняемые на компьютере. Основные классы задач подобного сорта перечислены ниже, в разделе о системах компьютерной алгебры. Имеется также специальное значение термина "компьютерная алгебра", применяемое к совокупности методов решения нелинейных систем алгебраических уравнений. В настоящей статье мы употребляем этот термин в широком смысле.

Следует отметить кардинальное различие между символьными вычислениями (т.е. методами и алгоритмами компьютерной алгебры) и численными методами. Вычисления с числами с плавающей точкой реализованы в компьютере на аппаратном уровне и, следовательно, выполняются с очень высокой скоростью. С другой стороны, представление вещественного (и даже рационального) числа, содержащего бесконечное количество цифр после десятичной точки, в виде числа с плавающей точкой, является приближенным. Поэтому главной заботой специалиста по численным методам и пользователя пакета для численных расчетов является контроль за точностью всех промежуточных вычислений, а его основной целью служит достижение заданной точности приближения для конечного результата. Уже вычисления с целыми числами произвольной разрядности не могут быть реализованы на аппаратном уровне и производятся на программном уровне. То же относится и к работе с рациональными числами, составленными из целых большой разрядности, и к полиномам

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований, грант 00-01-00441.

высокой степени, а также ко всем прочим объектам символьных вычислений. Следовательно, такие вычисления производятся с гораздо меньшей скоростью, чем расчеты с числами с плавающей точкой. Однако, поскольку символьные вычисления дают абсолютно точный результат, пользователь не должен заботиться о погрешности. Таким образом, каждый из указанных типов вычислений имеет свои преимущества и недостатки. Вопрос о том, какой из них предпочесть зависит, главным образом, от специфики решаемой задачи. Отметим также, что в последние годы становятся популярными и являются объектом пристального внимания исследователей так называемые гибридные методы, сочетающие методы компьютерной алгебры, численные методы, а также методы интервального анализа.

§2. Классификация систем и типы задач

За два последних десятилетия было создано огромное число систем компьютерной алгебры, большинство из которых широко применяется в научных вычислениях, при решении прикладных проблем, в индустрии. В зависимости от набора стандартных задач, которые могут быть решены при помощи данных систем, последние подразделяются на универсальные (или, иначе, системы общего назначения) и специализированные. Перечислим наиболее значимые типы задач, решаемые при помощи универсальных систем.

1. Всевозможные символьные вычисления с целыми числами произвольной длины, с рациональными числами без ограничения разрядности их числителя и знаменателя, с элементами конечных полей, с полями алгебраических чисел (т.е. с конечными расширениями поля Q, например, при помощи комплексных корней n-й степени из 1) и их кольцами целых.

2. Целый ряд задач по вычислениям с полиномами от нескольких переменных над указанными в предыдущем пункте полями и кольцами (нахождение наибольшего общего делителя, точного значения корней многочленов в радикалах, разложение на множители, вычисление дискриминантов и результантов, базисов Грёбнера и т.п.). Решение систем полиномиальных уравнений и других систем нелинейных уравнений, сводящихся к ним подстановками элементарных функций.

3. Символьные вычисления с полиномами над областью выра-

жений. В частности, вычисления с многочленами, коэффициенты которых содержат произвольное число параметров.

4. Символьное вычисление пределов функций, содержащих параметры.

5. Символьное дифференцирование, т.е. нахождение производных по стандартным правилам, изучаемым в высших учебных заведениях. Подчеркнем, что эти методы совершенно отличны от способов численного дифференцирования, известных в вычислительной математике.

6. Символьное интегрирование, т.е. вычисление неопределенных интегралов. Отметим коренное отличие соответствующих алгоритмов от численных методов, используемых для нахождения приближенных значений определенных интегралов.

7. Решение обыкновенных дифференциальных уравнений и систем дифференциальных уравнений в символьной форме.

8. Символьное решение уравнений математической физики и систем уравнений в частных производных.

9. В большинстве универсальных систем компьютерной алгебры имеются средства для изображения 2-мерной и 3-мерной графики. Таким образом, пользователь имеет возможность строить графики кривых и поверхностей, заданных параметрически. Как правило, можно управлять цветом, масштабом, разрешением и точкой наблюдения; окружать график осями, кубом и т.п.

В качестве примера упомянем несколько наиболее известных универсальных систем компьютерной алгебры: Maple, Mathematica, MuPAD, Axiom, Reduce, Derive, Macsyma, Maxima. К сожалению, большинство этих систем было разработано на коммерческой основе, поэтому их лицензионные версии продаются и стоят значительных денег. Приятное исключение составляет созданная в университете города Paderborn (Germany) система компьютерной алгебры MuPAD (Multi Processing Algebra Data Tool) [6]. Отметим также свободно распространяемую (на общих принципах лицензии GNU) универсальную систему Maxima, разработанную в техасском университете (Austin) У. Ф. Шелтером.

Из самого названия специализированных систем компьютерной алгебры ясно, что они специализированы на каком-либо типе задач или области. Как правило, такие системы и пакеты создавались группой специалистов-энтузиастов данного профиля на общественных началах в свободное от преподавательской работы время. Поэтому значительная часть таких специализированных систем распространяется разработчиками бесплатно для некоммерческих целей. Приведем несколько примеров таких систем и пакетов. Наиболее интересной и перспективной автору этих строк представляется система компьютерной алгебры GAP (Groups, Algorithms and Programming) [7], созданная в немецком городе Aachen и позже модифицированная в университете шотландского города St. Andrews. Изначально разрабатываемая как система для вычислений в группах, система GAP в настоящее время представляет собой мощный инструмент практически для любых дискретных алгебраических вычислений. В предпоследней версии GAP 3.4.4 имеется 26 подпакетов для эффективных вычислений с графами; кодами, исправляющими ошибки; матрицами больших размерностей; пространственными кристаллографическими группами и т.п. Последняя модификации GAP 4.2 включает в себя значительно более эффективные алгоритмы, многие из которых по быстродействию превосходят свои аналоги в GAP 3.4.4 в 2–3 раза. Обе версии системы обладают графическими средствами для изображения графов, соответствующих решеткам подгрупп заданной группы или каких-то подграфов данного графа в случае групп большого порядка. Эти возможности доступны лишь при использовании операционной системы UNIX (которая всегда предпочтительна для большинства систем компьютерной алгебры, несмотря на наличие версий для других операционных систем) и оболочки X-Windows. Система GAP распространяется свободно для некоммерческих целей.

Из других бесплатно распространяемых специализированных систем упомянем разработанную во Франции систему PARI-GP. Она предназначена для вычислений в теории чисел, содержит компактные и эффективные алгоритмы для символьных вычислений с алгебраическими числами, соответствующими кольцами и полиномами над этими кольцами. Системы имеет весьма малые размеры, умещается на двух 3.5” дискетах. Несколько систем компьютерной алгебры для вычислений в дискретной математике (с графами, кодами, дизайнами, таблицами меток, полиномами Шуберта и т.п.) было создано на кафедре Lehrstuhl 2 für Mathematik в университете немецкого города Bayreuth. Среди них бесплатно распространяемые пакеты Discreta, Symmetrica, а также используемая в химической промышленности система MOLGEN для перечисления всех изомеров органических соединений заданной структуры. Отметим также, что имеется огромное число систем компьютерной алгебры для вычисления в таких областях науки, как механика, астрономия, различные разделы физики и так далее. Мы не будем перечислять здесь названия и описывать возможности всех этих систем, а укажем для заинтересованного читателя адрес в Internet, где он сможет найти подробную информацию о системах компьютерной алгебры и другом математическом программном обеспечении. Страничка под названием “Mathematics Software” доступна через EMIS (Europe Mathematics Information Service) по адресу:

http://www.ras.ru/EMIS/PSU-mirror/MathLists/Software.html

§3. Особенности архитектуры и принципы построения

Теперь отметим ряд общих характеристик систем компьютерной алгебры. Большинство из них построено с использованием некоторых общих принципов в архитектуре. В первую очередь следует отметить, что большинство систем компьютерной алгебры изначально разрабатывалось для операционных систем типа UNIX, поэтому даже при наличии соответствующих модификаций для MS-DOS и Windows они наиболее эффективно работают в среде UNIX. Большая часть продвинутых (т.е. аккуратно спроектированных и достаточно мощных) систем имеет ядро, которое выполняет ряд ключевых функций. К последним относятся: автоматическое динамическое распределение памяти, ключевые операции с основными простыми и составными (обычно списки и записи) типами данных. Кроме того, большинство подобных систем имеет свой собственный язык программирования, обычно паскалевского типа, но, как правило, не требующий описания типов и объектов. Одной из важнейших функций ядра является интерпретация программ написанных на языке системы. В дополнение к этому, ядро содержит набор программных средств для тестирования, обнаружения синтаксических ошибок и т.п. В большинстве продвинутых систем компьютерной алгебры ядро написано на языке C, откуда и вытекает их наибольшая эффективность в среде UNIX.

Кроме ядра, хорошо разработанные системы компьютерной алгебры содержат библиотеки различных уровней. В первую очередь, это библиотеки функций и процедур, а также обширные библиотеки данных. Более того, известные продвинутые системы включают, как правило, различные подпакеты для эффективного решения каких-либо специальных задач. Подобная особенность объясняется тем, что в настоящее время для выполнения одного и того же типа символьных вычислений имеется несколько (порой даже несколько десятков) алгоритмов. Выбор того или иного алгоритма определяется ключевыми особенностями решаемой в данный момент задачи – например, степенью многочлена и числом его ненулевых коэффициентов; размерностью системы или матрицы и т.п. В ряде продвинутых систем, точнее в вышедших в последние годы их версиях, имеется возможность автоматического выбора алгоритма самой системой в зависимости от начальных данных. Например, такая возможность имеется в последней версии системы GAP 4.2.

Важной особенностью построения современных систем компьютерной алгебры является динамичность записей о каждом объекте. А именно, при выполнении всякого нового вычисления с данным объектом полученный результат фиксируется и хранится во вновь создаваемом поле соответствующей записи. Если пользователь захочет воспроизвести этот результат, то вместо повторного вычисления последний будет просто извлечен из соответствующей записи. Таким образом, за счет некоторого дополнительного расхода памяти экономится время работы процессора. Чем больше вычислений с данным объектом производит пользователь, тем больше компонент приобретает соответствующая запись. Естественно, что эта особенность имеет место только в рамках данного сеанса пользовательской работы с системой. Выход из системы уничтожает данные такого сорта. При решении очень емких задач этой особенностью можно воспользоваться для высвобождения памяти. С другой стороны, во многих системах (например, GAP) имеется возможность удалять целые записи и их отдельные компоненты вручную, не завершая сеанс работы.

К общим принципам построения современных систем для символьных вычислений можно отнести и наличие обширной документации, поставляемой в различных удобных для пользователя форматах. Во-первых, все подобные системы, даже такие небольшие, как система PARI-GP, имеют удобную помощь “on-line”, которая вызывается при помощи набора имени интересующего пользователя ключевого слова. Кроме того, дистрибутив обычно содержит “.html” файлы, удобные для чтения при помощи “netscape”, “Explorer”, “Opera” и т.п. В некоторых системах документация поставляется также в виде “.tex” или “.doc” файлов, позволяющих сформировать и распечатать полное руководство или справочник по системе. Для экономии ресурсов пользователю далеко не всегда имеет смысл распечатывать такое руководство, для большинства целей достаточно использовать помощь “on-line” непосредственно в процессе работы. Например, предпоследняя версия GAP 3.4.4 содержит руководство (Manual) объемом 1600 страниц в TeX.

Как уже упоминалось во введении, чистые системы компьютерной алгебры выполняют символьные вычисления. Однако в большинстве систем (особенно универсальных) для удобства пользователя реализованы числа с плавающей точкой и имеются функции для численной оценки полученного результата при подстановке значений параметров или при анализе выражений с радикалами. Например, в системе MuPAD для этой цели служит стандартная функция float.

Далее мы приведем два примера использования систем компьютерной алгебры для алгебраических вычислений: решение нелинейной алгебраической системы и вычисления с решетками подгрупп конечной группы с использованием таблицы меток Бернсайда.

§4. Примеры алгебраических вычислений

Как правило, выполнение прямых символьных вычислений не вызывает затруднения у пользователя универсальной или специализированной системы компьютерной алгебры. Обычно такие расчеты проводятся с использованием стандартных операторов `simplify` (упростить), `expand` (разложить, раскрыть скобки), `solve` (решить), `factorize` (разложить на множители) и т.д. Разумеется, синтаксис используемых команд зависит от конкретной системы, а имена стандартных функций и их аргументы могут быть уточнены в документации.

1°. Проиллюстрируем нахождение решений нелинейной алгебраической системы в пакете для символьных вычислений `Maxima` на примере системы с двумя неизвестными из работы [4]. Требуется решить следующую полиномиальную систему:

$$\begin{cases} 3x^2y + 2xy + y + 9x^2 + 5x - 3 = 0 \\ 2x^3y - xy - y + 6x^3 - 2x^2 - 3x + 3 = 0 \\ x^3y + x^2y + 3x^3 + 2x^2 = 0. \end{cases}$$

Использование всего одного оператора

```
solve([3*x*x*y+2*x*y+y+9*x+5*x-3=0,
2*x^3*y-x*y-y+6*x^3-2*x*x-3*x+3=0,
x^3*y+x*x*y+3*x^3+2*x*x=0],[x,y]);
```

приводит к получению всех трех решений исходной системы. Это

$$\left(-\frac{\sqrt{5}\cdot\sqrt{13}-5}{\sqrt{5}\cdot\sqrt{13}-9}, -\frac{\sqrt{5}\cdot\sqrt{13}+3}{4}\right);$$

$$(0,3);$$

$$\left(-\frac{\sqrt{5}\cdot\sqrt{13}+5}{\sqrt{5}\cdot\sqrt{13}+9}, \frac{\sqrt{5}\cdot\sqrt{13}-3}{4}\right).$$

После элементарного упрощения видим совпадения решений с результатами работы [4]:

$$(0,3);$$

$$((5+\sqrt{65})/4,\ -(3+\sqrt{65})/4);$$

$$((5-\sqrt{65})/4,\ (-3+\sqrt{65})/4).$$

Любопытно отметить, что от пользователя системы `Maxima`, которая позволила легко провести вычисления, не требуется знания базисов Гребнера, редуцированных базисов и прочих инструментов компьютерной алгебры. Вся эта техника уже заложена в алгоритмы данного пакета.

2°. Зачастую тормозом для решения конкретной задачи при помощи прямых символьных вычислений служит их значительная ресурсоемкость или низкая скорость реализованных алгоритмов. В такой ситуации решающее значение может иметь удачно подобранный для данной задачи метод вычисления. Так, автор этих строк в течение многих лет безуспешно пытался доказать гипотезу З. И. Боревича об эквивалентности понятий полинормальности и паранормальности в конечной разрешимой группе. Эти настойчивые попытки привели к мысли о несправедливости гипотезы и поиску контрпримера. Поскольку такой контрпример предполагал бы весьма тонкие различия в строении решеток промежуточных подгрупп, его построение теоретическими методами не представлялось возможным. С другой стороны, полный перебор всех подгрупп всех конечных групп до некоторого фиксированного порядка для поиска нужной решетки потребовал бы громадного времени ввиду крайне быстрого роста числа неизоморфных групп одного порядка и числа классов сопряженных подгрупп в них. Прямые вычисления с групповыми элементами и полный перебор явно не годились. В данной задаче помогла техника таблиц меток Бернсайда и разработанная в системе `GAP` [7] библиотека таблиц меток для подгрупп некоторых конечных простых групп.

При использовании этого инструмента медленные вычисления с групповыми элементами заменяются очень быстрыми операциями с целыми числами из таблицы меток при помощи данных из библиотеки `TOM` системы `GAP`. Подробности о методе, алгоритмах, теоремы о необходимых и достаточных условиях полинормальности и паранормальности и сам контрпример можно найти в работах автора [2, 3]. Здесь мы приведем лишь текст теста на свойство паранормальности для пояснения сказанного выше.

```
IsParaTom:=function(G,d)
local t, interm, comp, i, k, nk, pos1, pos2, nu1, nu2;
t:=TableOfMarks(G);
interm:=IntermClassesSubsTom(G,d);
# The author's functions is used here.
comp:=Filtered(interm, i->
        IsCompleteSubTom(G,d,i));
if not IsPolyTom(G,d)
then return false; fi;
for k in comp do
nk:=NormalizersTom(t)[k];
pos1:=Position(SubsTom(t)[k],d);
pos 2:=Position(SubsTom(t)[nk],d);
nu1:=NrSubsTom(t)[k][pos1];
```

```
  nu2:=NrSubsTom(t)[nk][pos2];
    if nu1 < > nu2 then return false; fi;
  od;
  return true;
end;
```

Литература

1. А. Акритас, *Основы компьютерной алгебры с приложениями.* М., Мир (1994), 544 с.
2. В. И. Мысовских, *Метки Бернсайда и решение двух проблем З. И. Боревича о полинормальных подгруппах.* Докл. Акад. наук, **367**, No. 4 (1999), 445–446.
3. В. И. Мысовских, *Субнормализаторы и свойства вложения подгрупп конечных групп.* Зап. научн. семин. ПОМИ **265** (1999), 258–280.
4. B. Buchberger, *Gröbner Bases: An Algorithmic Method in Polynomial Ideal Theory.* CAMP-Publ. Nr (1983), 83–290.
5. R. Fröberg, *An introduction to Gröbner bases.* Jonh Wiley and Sons (1997).
6. B. Fuchssteiner et al., *Multi processing algebra data tool: MuPAD.* Birkhäuser Verlag Basel (1994).
7. M. Schönert et al., *GAP – Groups, Algorithms and Programming.* Lehrstuhl D für Mathematik, RWTH Aachen, Germany, fifth edition (1995).

Mysovskikh V. I. Computer algebra packages and symbolic computation.

General principles for the construction and architecture of computer algebra packages are considered. The range of tasks that can be solved with their help is indicated. A classification of such systems is presented,and the most popular and significant packages for symbolic computing are listed. We mention fundamental characteristics of computer algebra systems, as well as data types. A difference between symbolic computation and numerical methods is indicated. Two examples of algebraic calculations in the packages `Maxima` and `GAP 4.2` are given. They concern the solution of nonlinear algebraic systems and the computations with subgroup lattices, respectively.

С.-Петербургский
государственный университет
e-mail:vimys@pdmi.ras.ru

Поступило 24 октября 2001 г.