БЫСТРЫЙ ПОИСК ДЕЛИТЕЛЯ ЖАНЕ*

В.П.Гердт, Д.А.Янович
Лаборатория вычислительной техники и автоматизации
Объединенный институт ядерных исследований
141980 Дубна, Московской обл.

Ю.А.Блинков
Механико-математический факультет
Саратовский университет
410071 Саратов

Представлен алгоритм быстрого поиска инволютивного мономиального делителя Жане. Такой поиск является важной составной частью построения мономиальных и полиномиальных базисов Жане. В качестве структуры данных для конечного множества мономов выбрано бинарное дерево, отражающего свойства деления Жане.

1 ВВЕДЕНИЕ

В работах [1, 2, 3] были разработаны общие алгоритмы построения полиномиальных инволютивных базисов для произвольных инволютивных делений и рассмотрен ряд конкретных делений. В работе [4] эти алгоритмы были обобщены на случай линейных систем дифференциальных уравнений в частных производных. Для деления Поммаре инволютивные полиномиальные алгоритмы ранее были реализованы в системе Reduce [1, 5, 6], а недавно общие полиномиальные и линейные дифференциальные алгоритмы были реализованы в системе Mathematica [7].

В настоящей работе мы представляем алгоритм и структуру данных для быстрого поиска делителя Жане среди элементов заданного конечного множества мономов. Поиск инволютивных делителей является одной из важнейших операций при построении инволютивных базисов и его эффективность во многом определяет эффективность инволютивных алгоритмов. Предлагаемый нами алгоритм опирается на специфические свойства деления Жане и наиболее адекватное ему представление множества мономов в виде бинарного дерева определенной структуры. Даны оценки временной сложности алгоритма в сравнении с алгоритмом бинарного поиска.

2 ДЕЛЕНИЕ ЖАНЕ

Ниже мы будем использовать следующие обозначения: $\mathbb{N}$ - множество неотрицательных целых чисел; $\mathbb{M} = \{x_1^{d_1} \cdots x_n^{d_n} \mid d_i \in \mathbb{N}\}$ - множество мономов в кольце $\mathbb{R} = \mathbb{K}[x_1, \ldots, x_n]$ многочленов над полем $\mathbb{K}$ нулевой характеристики; $deg_i(u)$ - степень переменной x_i в $u \in \mathbb{M}$; $deg(u) = \sum_{i=1}^{m} deg_i(u)$ - полная степень монома u; $\succ$ - допустимый порядок на мономах, который мы будем выбирать совместимым со следующим порядком переменных:

$$x_1 \succ x_2 \succ \cdots \succ x_n \,. \tag{1}$$

*Работа частично финансировалась за счет грантов РФФИ 98-01-00101, 00-15-96691 и грантов ИНТАС 96-0842, 99-1222.

Если моном u делит моном v мы будем записывать $u|v$.

Инволютивное мономиальное деление [1] задает для любого конечного множества мономов $U \subset \mathbb{M}$ и любого монома $u \in U$ разделение переменных на мультипликативные и немультипликативные Соответственно, $u \in U$ является инволютивным делителем монома $w \in \mathbb{M}$, если $u|w$ и моном w/u содержит только мультипликативные переменные для u.

К настоящему времени построен и изучен целый ряд инволютивных делений [1, 2, 3, 9]. В данной работе мы рассмотрим одно из этих делений, соответствующее классическому разделению переменных на мультипликативные и немультипликативные, введенному Жане для целей алгебраического исследования систем дифференциальных уравнений [10].

Определение 2.1. Деление Жане [1, 8]. Для каждого индекса переменной $1 \leq i \leq n$ распределим элементы U по подгруппам определяемым набором неотрицательных целых чисел $d_1, \ldots, d_i$:

$$[d_1, \ldots, d_i] = \{\ u\ \in U \mid d_j = deg_j(u),\ 1 \leq j \leq i\ \}. \tag{2}$$

Тогда переменная x_i является *мультипликативной по Жане* для $u \in U$ если $i = 1$ и $deg_1(u) = max\{deg_1(v) \mid v \in U\}$, или если $i > 1$, $u \in [d_1, \ldots, d_{i-1}]$ и $deg_i(u) = max\{deg_i(v) \mid v \in [d_1, \ldots, d_{i-1}]\}$. Если $u \in U$ является делителем Жане монома w, мы будем записывать это как $u|_J w$.

Для заданного конечного множества мономов U последующий алгоритм [3] строит минимальный базис Жане идеала $Id(U)$, порождаемого этим множеством.

Алгоритм: MonomialJanetBasis(U)

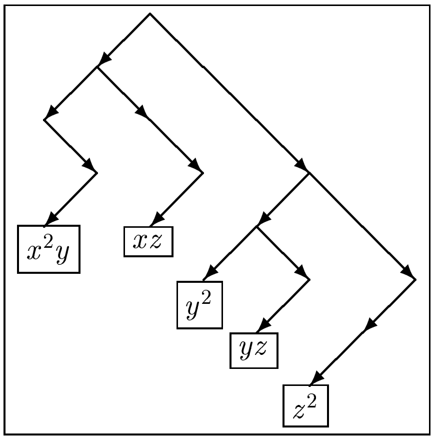
Input: U – множество мономов
Output: $\bar{U}$ – минимальный базис Жане идеала $Id(U)$
1: $\bar{U} := Autoreduce(U)$
2: **while** существует $u \in \bar{U}$ и $x \in NM_J(u, \bar{U})$, такие что $u \cdot x$ не имеет делителя Жане в $\bar{U}$ **do**
3: **choose** любое $\succ$ и пару u, x с наименьшим $u \cdot x$ по отношению к $\succ$
4: $\bar{U} := \bar{U} \cup \{u \cdot x\}$
5: **od**

На шаге 1 алгоритма выполняется авторедукция по обычной делимости, устраняющая из множества U все мономы, кратные другим мономам этого множества. На шаге 2 осуществляется поиск делителя Жане в текущем множестве мономов $\hat{U}$ для рассматриваемого немультипликативного продолжения $u \cdot x$. Если такой делитель отсутствует, то на шаге 4 происходит пополнение множества $\hat{U}$ данным продолжением.

3 АЛГОРИТМ ПОИСКА ДЕЛИТЕЛЯ ЖАНЕ

Ниже мы построим бинарное дерево, которое будем называть деревом Жане и структура которого отражает требуемое разбиение на подгруппы (2) множества U, упорядоченное внутри каждой подгруппы по степеням переменных. Мы покажем, что такая структура данных позволяет эффективно осуществлять поиск делителя Жане.

Прежде чем описывать общую структуру дерева поясним ее на конкретном примере. Рассмотрим множество мономов $U = \{x^2y, xz, y^2, yz, z^2\}$ $(x \succ y \succ z)$. и отобразим его в виде дерева Жане, как показано на следующем рисунке.



На данном рисунке листья дерева соответствуют мономам рассматриваемого множества. Левому потомку соответствует моном с увеличенной на единицу степенью текущей переменной. Правый потомок указывает на последующую переменную согласно выбранному упорядочению.

Рассмотрим теперь структуру бинарного дерева Жане (*JanetTree*, сокращенно JT) общего вида, как множества вершин и листьев $JT = \cup\{\nu\}$, сопоставляемого непустому множеству мономов. Каждой элементу дерева ν будем сопоставлять множество из четырех элементов $\nu = \{d,\, u,\, nd,\, nv\}$ со следующей структурой:

$$\begin{array}{rl} Deg(\nu) = d & \text{— степень переменной,} \\ Mon(\nu) = u & \text{— моном,} \\ Ndg(\nu) = nd & \text{— указатель на следующую по степени вершину,} \\ Nvr(\nu) = nv & \text{— указатель на следующую по переменной вершину.} \end{array}$$

При отсутствии следующей вершины, указателю на нее будем приписывать значение nil. Там где это не вызывает недоразумения, мы будем, для простоты, отождествлять указатели nd и nv с соответствующими им вершинами.

Корню дерева, который обозначим ν_o, мы сопоставим нулевую степень текущей переменной и единичный моном $u = 1$. Вершины и листья дерева JT обладают состояниями:

$$\begin{array}{rl} \text{Вершина:} & Nvr(\nu) \neq nil \vee (Ndg(\nu) \neq nil \wedge Deg(\nu) < Deg(Ndg(\nu))) \\ \text{Лист:} & Nvr(\nu) = nil \wedge Ndg(\nu) = nil \end{array}$$

Для быстрого поиска делителя Жане монома $w \neq 1$ в заданном дереве, соответствующем мономиальному множеству $U \neq \{1\}$, рассмотрим алгоритм **JanetDivisor** приведенный ниже.

Докажем корректность данного алгоритма. В соответствии с определением 2.1, делителем Жане монома w может быть только такой моном в U, который является старшим по лексикографическому упорядочению, соответствующему (1), среди всех мономов множества U, либо лексикографически старшим в одной из групп мономов (2) в U с $d_1 = deg_1(w), \ldots, d_i = deg_i(w)$ при некотором $1 \leq i < n$. При этом либо w должен совпадать с делителем, либо иметь степень по x_{i+1} большую, чем делитель.

Соответственно, корректность алгоритма **JanetDivisor** вытекает из того, что после окончания цикла 5-7 текущее состояние можно разделить на четыре альтернативных случая.

1. $Deg(\nu) = deg_i(w) \wedge Ndg \neq nil \wedge Nvr \neq nil$. В этом случае в U имеются мономы со степенью по x_i большей, чем степень этой переменной в w. Следовательно, надо продолжить поиск делителя среди мономов с такой же степенью по x_i, как и в w, переходя к следующей по порядку переменной, что осуществляется на шагах 10-12 алгоритма.

2. $Deg(\nu) < deg_i(w) \wedge Ndg(\nu) = nil \wedge Nvr \neq nil$. В этом случае опять-таки возможна делимость Жане по x_i и, как и в предыдущем случае, исследование делимости переносится на следующую переменную.

3. $Deg(\nu) = deg_i(m) \wedge Ndg(\nu) \neq nil \wedge Nvr = nil$. В этом случае делителя Жане нет. Это следует из того, что продолжение перехода влево приводит к листу дерева с мономом имеющим степень по текущей переменной, большую чем w. Такой моном не может, очевидно, делить w.

4. $Deg(\nu) \leq deg_i(m) \wedge Ndg(\nu) = nil \wedge Nvr = nil$. В этом случае ν является листом дерева, моном которого и есть искомый делитель Жане.

Алгоритм: JanetDivisor(JT, w)

Input: JT – дерево Жане, w – моном
Output: моном $u \,|_J\, w$ листа JT или nil в противном случае

```
d := deg(w)
i := 1
ν := ν_o
while d > 0 do
   while Deg(ν) < deg_i(w) and Ndg(ν) do
      ν := Ndg(ν)
   od
   if Nvr(ν) then
      d := d − deg_i(w)
      i := i + 1
      ν := Nvr(ν)
   elif Ndg(ν) then
      return nil
   else
      return Mon(ν)
   fi
od
```

Теорема. *Пусть d наибольшая полная степень мономов от n переменных из мономиального множества U. Тогда временная сложность алгоритма* **JanetDivisor** *и алгоритма бинарного поиска делителя Жане ограничены соответственно*

$$
\begin{aligned}
t_{\mathbf{JanetDivisor}(U)} &= O(d+n), \\
t_{\mathbf{BinarySearch}(U)} &= O(n((d+n)\log(d+n) - n\log(n) - d\log(d))).
\end{aligned}
$$

Доказательство. Временная сложность алгоритма **JanetDivisor** определяется наибольшей длиной пути в построенном бинарном дереве. Поскольку мономы ограничены полной степенью d, то d будет также является границей для длины пути если воспользоваться n-арным деревом. Поскольку алгоритм **JanetDivisor** использует бинарное дерево, время поиска монома будет ограничена $t_{\mathbf{JanetDivisor}(G)} = O(d+n)$.

Число различных мономов степени $\leq d$ от n переменных равно $\binom{d+n}{n}$. Временная сложность алгоритма бинарного поиска в упорядоченном множестве из k элементов равна $O(\log_2 k)$. Для количества элементов множества U имеет место ограничение $k \leq \binom{d+n}{n}$. Используя вариант формулы Стирлинга

$$\log(n!) - ((n+1/2)\log(n) - n) = O(1)$$

для оценки выражения

$$O(\log_2 \binom{d+n}{n}) = O\left(\log\left(\frac{(d+n)!}{n!d!}\right)\right)$$

и учитывая, что операция сравнения мономов, необходимая для работы алгоритма бинарного поиска, имеет временную сложность $O(n)$ получим

$$t_{\mathbf{JanetBinFind}(G)} = O(n((d+n)\log(d+n) - n\log(n) - d\log(d))).$$

$\square$

СПИСОК ЛИТЕРАТУРЫ

[1] *Gerdt V.P., Blinkov Yu.A.* Involutive Bases of Polynomial Ideals // Math. Comp. Sim. 1998, 45. P.519-542.

[2] *Gerdt V.P., Blinkov Yu.A.* Minimal Involutive Bases // Math. Comp. Simul. 1998, 45. P.543-560.

[3] *Gerdt V.P.* Involutive Division Technique: Some Generalizations and Optimizations // Записки научных семинаров СПОМИ (С.Петербург) 1999, 258. С.185-206.

[4] *Gerdt V.P.* Completion of Linear Differential Systems to Involution // Computer Algebra in Scientific Computing / CASC'99", V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 1999. P.115-137.

[5] *Жарков А.Ю., Блинков Ю.А.* Инволютивные системы алгебраических уравнений // Программирование. 1994, 1. С.53-56.

[6] *Zharkov A.Yu., Blinkov Yu.A.* Involutive Approach to Investigating Polynomial Systems // Math. Comp. Simul. 1996, 42. P.323-332.

[7] *Gerdt V.P., Kornyak V.V., Berth M., Czichowsky G.* Construction of Involutive Monomial Sets for Different Involutive Divisions // Computer Algebra in Scientific Computing / CASC'99", V.G.Ganzha, E.W.Mayr and E.V.Vorozhtsov (Eds.), Springer-Verlag, Berlin, 1999. P.147-157.

[8] *Гердт В.П., Блинков Ю.А.* Инволютивное деление мономов // Программирование. 1998, 6. С.22-24.

[9] *Chen Yu-Fu, Gao Xiao-Shan.* Vector Representation of Involutive Divisions // Mathematics-Mechanization Research Preprints, No.18, Beijing, 1999, C.9-22.

[10] *Janet M.* Leçons sur les Systèmes d'Equations aux Dérivées Partielles // Cahiers Scientifiques, IV, Gauthier-Villars, Paris, 1929.