ISSN 2079-3316 ПРОГРАММНЫЕ СИСТЕМЫ: ТЕОРИЯ И ПРИЛОЖЕНИЯ № 2(6), 2011, с. 97–103

УДК 517.951

С. Н. Тычков

Реализация скобки Кругликова–Лычагина–Майера в системе компьютерной алгебры Maple

Аннотация. В настоящей статье описана реализация скобки Кругликова–Лычагина–Майера на языке Maple. Результаты работы были анонсированы в [1] и [2].

Ключевые слова и фразы: дифференциальные уравнения, скобка Кругликова–Лычагина–Майера, Maple.

Введение

В работе предлагается реализация на языке Maple скобки Кругликова–Лычагина–Майера, которая используется для решения вопроса о формальной интегрируемости [3] системы дифференциальных уравнений. Пусть $\mathcal{E}$ — система r дифференциальных уравнений в частных производных:

$$(1)\qquad \mathcal{E}:\begin{cases} F_1\left(x, u, \dfrac{\partial u}{\partial x}, \dots, \dfrac{\partial^{|\sigma_1|}u}{\partial x^{\sigma_1}}\right) = 0,\\ F_2\left(x, u, \dfrac{\partial u}{\partial x}, \dots, \dfrac{\partial^{|\sigma_2|}u}{\partial x^{\sigma_2}}\right) = 0,\\ \quad\dots\\ F_r\left(x, u, \dfrac{\partial u}{\partial x}, \dots, \dfrac{\partial^{|\sigma_r|}u}{\partial x^{\sigma_r}}\right) = 0,\end{cases}$$

где $x = (x_1, x_2, \dots, x_n)$ — вектор размерности n, u — вещественная функция от x, $\sigma_i = ({\sigma_i}^1, {\sigma_i}^2, \dots, {\sigma_i}^n)$, $i = 1, 2, \dots, r$ — мультииндексы, элементы которых — неотрицательные целые числа. В этих обозначениях под $\partial^{|\alpha|}u/\partial x^\alpha$ мы понимаем частную производную

$$\frac{\partial^{|\alpha|}u}{\partial x_1^{\alpha^1}\partial x_2^{\alpha^2}\ldots\partial x_n^{\alpha^n}},$$

где α — мультииндекс, $|\alpha| = \alpha^1 + \ldots + \alpha^n$.

Для упрощения записи введем обозначение:

$$p_\sigma = \frac{\partial^{|\sigma|}u}{\partial x^\sigma}.$$

В координатах x, u,p_σ пространства джетов $\mathbf{J}^k\mathbb{R}^n$ $(k = \max\limits_i(|\sigma_i|))$ система (1) запишется следующим образом:

$$\left\{\begin{array}{l} F_1(x,u,\ldots,p_{\sigma_1}) = 0,\\ F_2(x,u,\ldots,p_{\sigma_2}) = 0,\\ \quad\ldots\\ F_r(x,u,\ldots,p_{\sigma_r}) = 0. \end{array}\right. \tag{2}$$

Под оператором полной производной по переменной x_j будем понимать оператор:

$$D_j = \frac{\partial}{\partial x_j} + \sum_\sigma p_{\sigma+1_j}\frac{\partial}{\partial p_\sigma}.$$

Определение 1. *Скобкой Кругликова–Лычагина–Майера (далее КЛМ-скобка) двух функций $F \in C^\infty(\mathbf{J}^k\mathbb{R}^n)$ и $G \in C^\infty(\mathbf{J}^l\mathbb{R}^n)$ называется функция*

$$[F,G] \stackrel{def}{=} \sum_{|\alpha|=k}\frac{\partial F}{\partial p_\alpha}D^\alpha G - \sum_{|\beta|=l}\frac{\partial G}{\partial p_\beta}D^\beta F, \tag{3}$$

где $D^\alpha = D_1^{\alpha^1}\circ\ldots\circ D_n^{\alpha^n}$, $\alpha = (\alpha^1,\ldots,\alpha^n)$, а D_i — оператор полной производной по переменной x_i $(i = 1\ldots,n)$.

Система (1) определяет подмногообразие $\mathcal{E}$ в пространстве джетов, определенное формулой (2). Скобка КЛМ представляет собой функцию на пространстве джетов. Ограничение этой функции на подмногообразие $\mathcal{E}$ будем называть *скобкой в силу системы (2)*.

Замечание 1. *Алгоритм исследования формальной интегрируемости основывается на следующем результате. Если система (2) формально интегрируема, то КЛМ-скобки*

$$[F_i,F_j] \quad (i,j = 1,\ldots,r, i \neq j)$$

равны нулю в силу системы $\mathcal{E}$. Обратно, если все скобки Майера равны нулю в силу $\mathcal{E}$, а характеристический идеал [3] $\{F_1, F_2, \ldots, F_r\}$ есть полное пересечение [4], то система формально интегрируема. Исчерпывающее описание данного метода с доказательством можно найти в [5].

В программах для символьных вычислений Maple и Mathematica есть пакеты для решения обыкновенных дифференциальных уравнений, так и в частных производных.

В частности, в Maple представлен пакет *diffalg* для исследования полиномиальных систем дифференциальных уравнений в частных производных. В этом пакете с помощью базисов Грёбнера можно исследовать совместность переопределенных систем.

В работе [6] проведено сравнение методов дифференциальных базисов Грёбнера и КЛМ-скобки.

Отметим, что процедура вычисления скобки Кругликова–Лычагина–Майера не была ранее реализована ни в одном пакете символьных вычислений.

1. Вспомогательные процедуры

Для реализации формулы (3) нам прежде всего необходима процедура вычисления полной производной по независимой переменной.

Удобно записывать частную производную функции n переменных $\partial^\alpha f/\partial x^\alpha$ в виде координат джетов f_α. Чтобы в нашей программе предусмотреть возможность записывать производные функций в таком виде, заведём два списка `ivars` и `dvars`, содержащие имена независимых и зависимых переменных соответственно. Например, пусть заданы `ivars := [x, y]` и `dvars := [u, v]`. Тогда под записью `u[1, 2]` мы понимаем производную u_{xyy}.

Также объявим переменные `ivarsCount` и `dvarsCount`, хранящие количества элементов в списках `ivars` и `dvars` соответственно.

Реализация вычисления оператора полной производной опирается на обычные правила дифференцирования. Заметим, что нужно поддерживать операцию дифференцирования над записаннными в виде джетов производными зависимых переменных (`dvars`). Ниже представлен исходный код функции `TDiff`, реализующей вычисление полной производной.

```
TDiff := proc( expr, varIndex::integer, $ )
 local m_1_, m_2_;
 if varIndex <= 0 or varIndex > ivarsCount then
  error "Incorrect variable index: %1", varIndex;
 end if;
 if not has( expr, dvars ) then
  return diff( expr, ivars[varIndex] );
 elif type( expr, '`+`' ) then
  return map( procname, args );
 elif type( expr, '`*`' ) then
  m_1_, m_2_ := op( 1, expr ), subsop( 1 = 1, expr );
  return procname( m_1_, varIndex ) * m_2_ +
  m_1_ * procname( m_2_, varIndex );
 elif type( expr, '`^`' ) then
  m_1_, m_2_ := op( expr );
  return expr * ( TDiff( m_2_, varIndex ) * ln( m_1_ ) +
                  TDiff( m_1_, varIndex ) * m_2_ / m_1_ );
 elif type(expr, `function`) then
  m_1_ := 0;
  for m_2_ from 1 to nops( expr ) do
   m_1_ := m_1_ + D[m_2_]( op( 0, expr ) )( op( expr ) ) *
                           TDiff( op( m_2_, expr ), varIndex );
  end do;
  return m_1_;
 elif checkJet( expr ) then
  m_1_ := op( 0, expr );
  m_2_ := op( varIndex, expr ) + 1;
  return m_1_ [op( subsop( varIndex = m_2_, [op( expr )] ) )];
 else
  error "Unexpected expression to find derivative of %1 by %2",
  expr, varIndex;
 end if;
end proc:
```

В качестве первого аргумента этой функции может выступать любое выражение, содержащее независимые и зависимые переменные, второй аргумент — целое число от 1 до значения переменной `ivarsCount`, выражающее индекс переменной, по которой дифференцируется выражение.

Для непосредственных вычислений мы создадим функцию-обертку `TotDiff` (не следует путать со встроенной функцией `TotalDiff` пакета `JetCalculus`), первый аргумент которой дифференцируемое выражение, а второй — список целых индексов переменных дифференцирования.

```
TotDiff := proc( expr, vars::list )
```

```
 local varIndex::integer, result;
  result := expr;
  for varIndex in vars do
   result := TDiff( result, varIndex );
  end do;
  return result;
end proc;
```

2. Реализация

Ниже представлена реализация формулы (3) на языке Maple.

```
MayerBracket := proc(F, G, f::symbol)
local alphas::list, betas::list, sigma::list,
      orderF::integer, orderG::integer, result;

 orderF := EquationOrder( F, f );
 orderG := EquationOrder( G, f );
 alphas := select( x -> evalb( `+`( op ( x ) ) = orderF ),
                  allSigmas( F, f ) );
 betas := select( x -> evalb( `+`( op ( x ) ) = orderG ),
                 allSigmas( G, f ) );
 result := 0;
 for sigma in alphas do
  result := result + diff( F, f[op( sigma )] ) *
                 TotDiff( G, sigma2vars( sigma ) );
 end do;
 for sigma in betas do
  result := result - diff( G, f[op( sigma )] )
               * TotDiff( F, sigma2vars( sigma ) );
 end do;
 return simplify( result );
end proc;
```

3. Пример

Рассмотрим переопределенную систему уравнений из работы [6]:

$$\begin{cases} F_1 = u_{yyy} + u_{xxx}u_{xyy} - u_{xxy}^2 = 0, \\ F_2 = u_{xx}u_{yy} - cu_{xy}^2 = 0, \end{cases} \tag{4}$$

где $c \in \mathrm{R}$. Наша программа для вычисления КЛМ-скобки этой системы представлена ниже. Комментарии к инструкциям приведены в самой программе.

```
#импортирует нужные процедуры из пакета Brackets
with(Brackets):
 #задает списки независимых и зависмых переменных
```

```
setup([x, y], [u]):
#первое уравнение
F[1] := u[0, 3] + u[3, 0]*u[1, 2] - u[2, 1]^2:
#второе уравнение
F[2] := u[2, 0]*u[0, 2] - c*u[1, 1]^2:
#вычисляем скобку Майера
F[3] := MayerBracket(F[2], F[1], u):
#продолжение системы в 4-джеты
D4 := {TotDiff(F[1], [1]), TotDiff(F[1], [2]), TotDiff(F[2], [1, 1]),
TotDiff(F[2], [1, 2]), TotDiff(F[2], [2, 2])}:
#выражение производных 4-го порядка через производные меньших порядков
Der4 := solve(D4, {u[0, 4], u[1, 3], u[2, 2], u[3, 1], u[4, 0]}):
#продолжение системы в 3-джеты
D3 := {F[1], TotDiff(F[2], [1]), TotDiff(F[2], [2])}:
#выражение производных 4-го порядка через производные меньших порядков
Der3 := solve(D3, {u[0, 3], u[1, 2], u[3, 0]}):
#исключаем из F_3 четвертые и третьи производные
Fm := factor(simplify(eval(eval(F[3], Der4), Der3))):
```

В результате работы этой программы мы получаем КЛМ-скобку для системы (4):

$$[F_1, F_2] = \frac{4}{u_{xx}^7}\left(\frac{3}{2} - c\right)\left(2T^2 + 4cu_{xy}\sqrt{RS}T - u_{xx}^4 u_{yy}^2 u_{xxy}^2\right),$$

где

$$R = u_{yy} + u_{xx}u_{xxy},$$
$$S = Rc^2u_{xy}^2 - u_{xx}^2 u_{yy} u_{xxy},$$
$$T = S + Rc^2u_{xy}^2.$$

Скобка $[F_1, F_2]$ тождественно равна нулю, если и только если $c = \frac{3}{2}$.

Список литературы

[1] Тычков С. Н. *Вычисление скобки Кругликова-Лычагина-Майера в системе Maple* // Международная конференция «Геометрия в Одессе 2009» (25–30 мая 2009). — Одесса : ВПП «Друкарський дім», с. 76. ↑[]

[2] Тычков С. Н. *Реализация скобки Кругликова–Лычагина–Майера в системах символьной математики* // Международная конференция «Геометрия в Астрахани 2009» (10–16 сентября 2009). — Астрахань : Издательский дом «Астраханский университет», с. 31. ↑[]

[3] Алексеевский Д. В., Виноградов А. М. Основные идеи и понятия дифференциальной геометрии. М. : ВИНИТИ, 1988. — 289 с. ↑[], 1

[4] Кокс Д., Литтл Д., О'Ши Д. Идеалы, многообразия и алгоритмы. М. : Мир, 2000. — 687 с. ↑1

[5] Lychagin V., Kruglikov B. *Mayer brackets and solvability of PDEs—I* // Differential Geometry and its Applications, 2002. **17**, p. 251–272. ↑1

[6] Kruglikov B. *Note on two compatibilty criteria: Jacobi-Mayer bracket vs. differential Groebner basis* // Lobachevskii Journal of Mathematics, 2006. **23**, p. 57–70. ↑[], 3

S. N. Tychkov. *Implementation of Kruglikov–Lychagin–Mayer bracket on Maple.*

Abstract. In this article we describe an implementation of Kruglikov–Lychagin–Mayer bracket on Maple.

Key Words and Phrases: differential equations, Kruglikov–Lychagin–Mayer bracket, Maple.

Образец ссылки на статью:

С. Н. Тычков. *Реализация скобки Кругликова–Лычагина–Майера в системе компьютерной алгебры Maple* // Программные системы: теория и приложения : электрон. научн. журн. 2011. № 2(6), с. 97–103. URL: http://psta.psiras.ru/read/psta2011_2_97-103.pdf