

Обращение преобразования Миуры

В.Э. Адлер. Классические интегрируемые системы, весенний семестр 2020

В этом файле показано, как можно реализовать символьные вычисления, связанные с оператором полной производной, эволюционным дифференцированием, вариационной производной и алгоритмом интегрирования по частям. Это иллюстрируется на примере задачи о законах сохранения КдФ; здесь мы будем строить их при помощи преобразования Миуры. Определённые в этом файле функции неоднократно пригодятся и в дальнейшем.

1 Что такое символьные вычисления

Есть несколько уровней работы с дифференциальными уравнениями. При численном построении решения (например, по схеме Забуски-Краскала из первой лекции) обрабатываются массивы вещественных чисел. Тут без компьютера никак не обойтись.

При построении точных решений типа солитонных компьютер тоже помогает – делает вычисления с такими функциями как, например, $\cosh x$ и рисует графики.

Есть и более абстрактные вычисления. Пусть задано некоторое ДУ на функцию $u(x, t)$. Можно работать с этой функцией и её производными просто как с некими символами, формальными переменными. При таком подходе мы фактически работаем не с решениями уравнения, а с самим уравнением; всё, что у нас есть – это оно само и его дифференциальные следствия, то есть, соотношения, которые получаются при дифференцировании по x и t (в предположении, что $u \in C^\infty$). В теории интегрируемых систем вычисления этого типа встречаются очень часто и играют важную роль. В принципе, тут некоторое время можно обходиться без компьютера. Чтобы как следует понять суть вычислений, которые у нас будут, я советую некоторую их часть делать на бумаге. Например, ниже приведен алгоритм построения законов сохранения для уравнения КдФ; попробуйте несколько первых законов сохранения найти и проверить вручную. Через несколько лекций будут похожие вычисления с высшими симметриями, тоже несколько первых нетрудно найти вручную. Вы увидите, что выкладки простые, но их объем быстро растет в зависимости от порядка по производным. Компьютер просто помогает сделать на несколько шагов больше.

2 Основные операции

Рассмотрим эволюционные уравнения с одной полевой переменной $u = u_0$:

$$(1) \quad u_t = a(x, u_0, u_1, \dots, u_n), \quad u_n := \partial_x^n(u).$$

Основная идея заключается в том, чтобы считать x , t и все u_n независимыми (“динамическими”) переменными. Мы рассматриваем всевозможные (гладкие) функции от конечного числа этих переменных – например, саму правую часть a , правые части высших симметрий, плотности законов сохранения; в общем, всё что угодно. То обстоятельство, что u_n имеют смысл производных от u , кодируется в правилах дифференцирования таких функций. Чтобы их задать, достаточно определить следующие функции (не забудьте активировать эту ячейку, для этого нужно поместить на неё курсор и нажать Shift-Enter):

```

In[1]:= vars[f_] := Union[Cases[f, _u, {0, ∞}]]
dif[f_] := Plus @@ (D[f, #] dd[#] & /@ vars[f])
dx[f_] := D[f, x] + dif[f] /. dd[u[k_]] => u[k + 1]
dx[f_, k_] := Nest[dx, f, k]
star[g_, f_] := dif[g] /. dd[u[k_]] => dx[f, k]
vard[f_] := Expand[Plus @@ ((-1) ^ # [1] dx[D[f, #], # [1]] & /@ vars[f])]

```

Остальные ячейки в этом разделе можно не запускать, они служат лишь для тестирования и пояснения того, что означают эти определения.

Функция `vars[f]` возвращает список переменных u_n в выражении f (если мы применяем его не упростив f , то этот список может оказаться более широким, чем нужно, это не страшно):

```

In[7]:= f = (u[1] + u[13])^2 - u[1]^2 - 2 u[1] u[13] + b[x]
vars[f]
Expand[f]
vars[%]

```

```
Out[7]= b[x] - u[1]^2 - 2 u[1] u[13] + (u[1] + u[13])^2
```

```
Out[8]= {u[1], u[13]}
```

```
Out[9]= b[x] + u[13]^2
```

```
Out[10]= {u[13]}
```

Функция `dif[f]` возвращает дифференциал выражения, причем $d u_n$ обозначается `dd[u[n]]`. К самим функциям `dif` и `dd` мы обращаться не будем, но они используются в определении функции `dx[f]`, которая возвращает полную производную

$$D_x(f) = \partial_x(f) + u_1 \partial_0(f) + \dots + u_{n+1} \partial_n(f),$$

где $f = f(x, u_0, u_1, \dots, u_n)$ и ∂_k обозначает частную производную по u_k . Обратим внимание, что мы не пытаемся упростить выражение по ходу вычислений. Практика показывает, что это лучше делать в конце, при помощи подходящей команды (`Expand`, `Together`, `Factor`, `Collect`, `Simplify`, ...). Если окончательный результат хочется увидеть в индексных обозначениях, это тоже можно сделать (но в процессе вычислений использовать индексы неудобно).

```

In[11]:= f
dif[f]
dx[f]
Expand[dx[f]]
Expand[dx[f, 2]]
% /. u[n_] => u_n

```

```
Out[11]= b[x] - u[1]^2 - 2 u[1] u[13] + (u[1] + u[13])^2
```

```
Out[12]= dd[u[13]] (-2 u[1] + 2 (u[1] + u[13])) + dd[u[1]] (-2 u[1] - 2 u[13] + 2 (u[1] + u[13]))
```

```
Out[13]= u[2] (-2 u[1] - 2 u[13] + 2 (u[1] + u[13])) + (-2 u[1] + 2 (u[1] + u[13])) u[14] + b'[x]
```

```
Out[14]= 2 u[13] u[14] + b'[x]
```

```
Out[15]= 2 u[14]^2 + 2 u[13] u[15] + b''[x]
```

```
Out[16]= 2 u[14]^2 + 2 u[13] u[15] + b''[x]
```

Функция `star[g, f]` возвращает

$$g_*(f) = \nabla_f(g) = f \partial_0(g) + \dots + D_x^m(f) \partial_m(g),$$

где $g = g(x, u_0, u_1, \dots, u_m)$, то есть, результат дифференцирования выражения g в силу уравнения $u_t = f$.

```
In[17]:= star[u[0], u[3] + 6 u[0] u[1]]
star[u[1], u[3] + 6 u[0] u[1]]
star[u[0] u[2], u[3] + 6 u[0] u[1]]
```

```
Out[17]= 6 u[0] u[1] + u[3]
```

```
Out[18]= 6 u[1]^2 + 6 u[0] u[2] + u[4]
```

```
Out[19]= u[2] (6 u[0] u[1] + u[3]) + u[0] (18 u[1] u[2] + 6 u[0] u[3] + u[5])
```

Функция `vard[f]` возвращает вариационную производную (= оператор Эйлера) от f :

$$\delta f / \delta u = E(f) = \partial_0(f) - D_x(\partial_1(f)) + D_x^2(\partial_2(f)) - \dots + (-D_x)^n(\partial_n(f)),$$

где $f = f(x, u_0, u_1, \dots, u_n)$ и ∂_k обозначает частную производную по u_k .

```
In[20]:= vard[u[3] + 6 u[0] u[1]]
vard[u[2] g[u[0]]]
```

```
Out[20]= 0
```

```
Out[21]= 2 u[2] g'[u[0]] + u[1]^2 g''[u[0]]
```

Теорема. Выражение является полной производной по x тогда и только тогда, когда его вариационная производная равна 0:

$$E(f) = 0 \Leftrightarrow \exists g : f = D_x(g).$$

Напишите здесь любое выражение вместо f (ну, не очень сложное, такое, чтобы его смогла осилить машина!) и проверьте, что в последней строчке получится 0.

```
In[22]:= f = g[u[3] + u[2]] u[4]^2 Sin[x] Cos[u[2]]
dx[f]
Expand[vard[%]]
```

```
Out[22]= Cos[u[2]] g[u[2] + u[3]] Sin[x] u[4]^2
```

```
Out[23]= Cos[x] Cos[u[2]] g[u[2] + u[3]] u[4]^2 +
2 Cos[u[2]] g[u[2] + u[3]] Sin[x] u[4] u[5] + Cos[u[2]] Sin[x] u[4]^3 g'[u[2] + u[3]] +
u[3] (-g[u[2] + u[3]] Sin[x] Sin[u[2]] u[4]^2 + Cos[u[2]] Sin[x] u[4]^2 g'[u[2] + u[3]])
```

```
Out[24]= 0
```

3 Преобразование Миуры

Рассмотрим модифицированное уравнение КдФ, точнее, мКдФ⁻:

$$(2) \quad v_t = v_{xxx} - 6(v^2 + \lambda)v_x,$$

с произвольным параметром λ . Определим переменную

$$(3) \quad u = -v_x - v^2 - \lambda.$$

Дифференцируя u по x и по t , с заменой v_t из (2), нетрудно проверить (сделайте это вручную), что u удовлетворяет уравнению КдФ

$$(4) \quad u_t = u_{xxx} + 6u u_x.$$

Чтобы проверить это с помощью определенных выше операций, мы временно поменяем ролями u и v (так как правила дифференцирования у нас заданы для u):

```
In[25]:= ut = u[3] - 6 (u[0]^2 + λ) u[1];
v = -u[1] - u[0]^2 - λ;
-star[v, ut] + dx[v, 3] + 6 v dx[v]
Expand[%]

Out[27]= -12 u[0] u[1]^2 + 6 (-λ - u[0]^2 - u[1]) (-2 u[0] u[1] - u[2]) -
6 (λ + u[0]^2) u[2] - 6 u[1] u[2] - 2 u[0] u[3] + 2 u[0] (-6 (λ + u[0]^2) u[1] + u[3])

Out[28]= 0
```

Замечание. У мКдФ есть две вещественные формы, отличающиеся знаком нелинейного члена:

$$v_t = v_{xxx} \pm 6(v^2 + \lambda)v_x.$$

Очевидно, они связаны заменой $v \rightarrow i v$, $\lambda \rightarrow -\lambda$, но она не годится, если мы следим за вещественностью решений. С этой точки зрения, свойства этих двух уравнений довольно сильно отличаются (например, у мКдФ⁺ есть решения в виде бризеров, а у мКдФ⁻ нет; представления нулевой кривизны имеют разный вид). Вещественное преобразование Миуры (3) определено только для мКдФ⁻. Однако, сейчас это для нас не важно, и вместо (2), (3) мы могли бы рассматривать уравнения

$$v_t = v_{xxx} + 6(v^2 + \lambda)v_x, \quad u = -i v_x + v^2 + \lambda:$$

```
In[29]:= ut = u[3] + 6 (u[0]^2 + λ) u[1];
v = -i u[1] + u[0]^2 + λ;
Expand[-star[v, ut] + dx[v, 3] + 6 v dx[v]]

Out[31]= 0
```

Преобразование (3) это пример так называемых *дифференциальных подстановок*. В отличие от *точечных замен переменных* (таких, как растяжение переменных, переход в подвижную систему координат, переход от декартовых координат к полярным и т.п.) это преобразование необратимо: нет такой формулы, которая выражает v как функцию от конечного числа переменных u_n . Тем не менее, такие односторонние преобразования тоже могут быть очень полезными. Во-первых, иногда уравнение может быть *линеаризовано* с их помощью. Вот простой пример, подстановка Коула-Хопфа. Здесь уравнение для u – линейное уравнение теплопроводности, и по любому его решению мы строим решение v нелинейного уравнения Бюргерса (тоже проверьте на бумаге, это совсем просто):

```
In[32]:= ut = u[2];
v = u[1] / u[0];
Expand[-star[v, ut] + dx[v, 2] + 2 v dx[v]]

Out[34]= 0
```

Во-вторых, даже если подстановка не приводит к упрощению уравнения (непонятно, чем (2) лучше чем (4)), её, возможно, удастся как-то использовать для построения частных решений. Действительно, на одной из следующих лекций мы увидим, что преобразования Миуры можно соединить в цепочку, позволяющую строить многосолитонные решения.

В-третьих, прямо сейчас мы увидим, что из формулы (3) можно очень просто получить рекуррентную схему для вычисления законов сохранения уравнения КдФ. Подобный приём срабатывает и во многих других примерах.

4 Обращение преобразования Миуры

Хотя выше было сказано, что преобразование (3) необратимо, мы всё же можем выразить v в виде ряда по обратным степеням параметра $(-\lambda)^{1/2}$, причём коэффициенты ряда будут многочленами от u_n . Сходимость ряда нас не интересует – это просто формальный ряд, производящая функция. Положим $\lambda = -z^2$ и перепишем преобразование (3) в виде

$$(5) \quad v^2 + v_x = z^2 - u.$$

Подставим сюда разложение

$$(6) \quad v = -z + V_0 + V_1/z + V_2/z^2 + \dots$$

(индекс у V_k это просто номер, не производная), это даст

$$\begin{aligned} z^2 - 2zV_0 + (-2V_1 + V_0^2) + (-2V_2 + 2V_0V_1)/z \\ + (-2V_3 + 2V_0V_2 + V_1^2)/z^2 + \dots + V_{0,x} + V_{1,x}/z + V_{2,x}/z^2 + \dots = z^2 - u. \end{aligned}$$

Отсюда следует, что $V_0 = 0$, $V_1 = u/2$, а на остальные коэффициенты получаем рекуррентные соотношения

$$2V_{n+1} = V_1V_{n-1} + \dots + V_{n-1}V_1 + V_{n,x} = 0, \quad n = 1, 2, 3, \dots$$

Вычислим несколько первых коэффициентов (скажем, 13, хотя машина может гораздо больше; на бумаге продублируйте, пока не надоест).

In[35]:=

```
M = 13;
V[1] = u[0] / 2;
Do[V[n + 1] = Factor[ (dx[V[n]] + Sum[V[s] V[n - s], {s, 1, n - 1}]) / 2], {n, 1, M - 1}];
```

Чтобы было удобнее смотреть, заменим $u[n]$ на u_n и умножим на степень 2:

In[38]:=

```
Do[Print[Vn, " = ", 2^n V[n] /. u[k_] => u_k], {n, 1, M}]
```

$$V_1 = u_0$$

$$V_2 = u_1$$

$$V_3 = u_0^2 + u_2$$

$$V_4 = 4 u_0 u_1 + u_3$$

$$V_5 = 2 u_0^3 + 5 u_1^2 + 6 u_0 u_2 + u_4$$

$$V_6 = 16 u_0^2 u_1 + 18 u_1 u_2 + 8 u_0 u_3 + u_5$$

$$V_7 = 5 u_0^4 + 50 u_0 u_1^2 + 30 u_0^2 u_2 + 19 u_2^2 + 28 u_1 u_3 + 10 u_0 u_4 + u_6$$

$$V_8 = 64 u_0^3 u_1 + 60 u_1^3 + 216 u_0 u_1 u_2 + 48 u_0^2 u_3 + 68 u_2 u_3 + 40 u_1 u_4 + 12 u_0 u_5 + u_7$$

$$V_9 = 14 u_0^5 + 350 u_0^2 u_1^2 + 140 u_0^3 u_2 + 442 u_1^2 u_2 + 266 u_0 u_2^2 + 392 u_0 u_1 u_3 + 69 u_3^2 + 70 u_0^2 u_4 + 110 u_2 u_4 + 54 u_1 u_5 + 14 u_0 u_6 + u_8$$

$$V_{10} = 256 u_0^4 u_1 + 960 u_0 u_1^3 + 1728 u_0^2 u_1 u_2 + 1224 u_1 u_2^2 + 256 u_0^3 u_3 + 900 u_1^2 u_3 + 1088 u_0 u_2 u_3 + 640 u_0 u_1 u_4 + 250 u_3 u_4 + 96 u_0^2 u_5 + 166 u_2 u_5 + 70 u_1 u_6 + 16 u_0 u_7 + u_9$$

$$V_{11} = 42 u_0^6 + 2100 u_0^3 u_1^2 + 1105 u_1^4 + 630 u_0^4 u_2 + 7956 u_0 u_1^2 u_2 + 2394 u_0^2 u_2^2 + 1262 u_2^3 + 3528 u_0^2 u_1 u_3 + 5564 u_1 u_2 u_3 + 1242 u_0 u_3^2 + 420 u_0^3 u_4 + 1630 u_1^2 u_4 + 1980 u_0 u_2 u_4 + 251 u_4^2 + 972 u_0 u_1 u_5 + 418 u_3 u_5 + 126 u_0^2 u_6 + 238 u_2 u_6 + 88 u_1 u_7 + 18 u_0 u_8 + u_{10}$$

$$V_{12} = 1024 u_0^5 u_1 + 9600 u_0^2 u_1^3 + 11520 u_0^3 u_1 u_2 + 13560 u_1^3 u_2 + 24480 u_0 u_1 u_2^2 + 1280 u_0^4 u_3 + 18000 u_0 u_1^2 u_3 + 10880 u_0^2 u_2 u_3 + 9524 u_2^2 u_3 + 7000 u_1 u_3^2 + 6400 u_0^2 u_1 u_4 + 11140 u_1 u_2 u_4 + 5000 u_0 u_3 u_4 + 640 u_0^3 u_5 + 2720 u_1^2 u_5 + 3320 u_0 u_2 u_5 + 922 u_4 u_5 + 1400 u_0 u_1 u_6 + 658 u_3 u_6 + 160 u_0^2 u_7 + 328 u_2 u_7 + 108 u_1 u_8 + 20 u_0 u_9 + u_{11}$$

$$V_{13} = 132 u_0^7 + 11550 u_0^4 u_1^2 + 24310 u_0 u_1^4 + 2772 u_0^5 u_2 + 87516 u_0^2 u_1^2 u_2 + 17556 u_0^3 u_2^2 + 69006 u_1^2 u_2^2 + 27764 u_0 u_2^3 + 25872 u_0^3 u_1 u_3 + 33760 u_1^3 u_3 + 122408 u_0 u_1 u_2 u_3 + 13662 u_0^2 u_3^2 + 26322 u_2 u_3^2 + 2310 u_0^4 u_4 + 35860 u_0 u_1^2 u_4 + 21780 u_0^2 u_2 u_4 + 20922 u_2^2 u_4 + 30776 u_1 u_3 u_4 + 5522 u_0 u_4^2 + 10692 u_0^2 u_1 u_5 + 20376 u_1 u_2 u_5 + 9196 u_0 u_3 u_5 + 923 u_5^2 + 924 u_0^3 u_6 + 4270 u_1^2 u_6 + 5236 u_0 u_2 u_6 + 1582 u_4 u_6 + 1936 u_0 u_1 u_7 + 988 u_3 u_7 + 198 u_0^2 u_8 + 438 u_2 u_8 + 130 u_1 u_9 + 22 u_0 u_{10} + u_{12}$$

Можно проверить, что построенный отрезок ряда действительно удовлетворяет уравнению (5), с точностью до членов z^n с $n \geq M$:

```
In[39]:= v = -z + Sum[V[n]/z^n, {n, 1, M}];
Series[Expand[v^2 + dx[v]], {z, Infinity, M - 1}]
```

```
Out[40]= z^2 - u[0] + O[1/z]^13
```

Прекрасно, и зачем это было нужно? А вот зачем: оказывается, что многочлены V_n являются *плотностями* законов сохранения для КдФ, то есть, существуют некоторые функции s_n от u_k такие, что

$$(7) \quad D_t(V_n) = D_x(s_n).$$

Эти функции мы пока не знаем, но можем убедиться, что они есть, применив к этому равенству вариационную производную, тогда правая часть уйдёт в 0. Проверяем:

```
In[41]:= Table[
  varD[star[V[n], u[3] + 6 u[0] u[1]]],
  {n, 1, M}]
```

```
Out[41]= {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```

Следует сделать одну оговорку. Если посмотреть внимательно на выписанную выше таблицу, можно заметить, что все V_n с четными номерами сами являются полными производными по x , ещё до дифференцирования по t :

In[42]:= Table[**vard**[V[n]], {n, 2, M, 2}]

Out[42]= {0, 0, 0, 0, 0, 0}

Такие плотности не засчитываются, они отвечают *тривиальным* законам сохранения $D_t(D_x(a)) = D_x(D_t(a))$, которые выполняются для любой функции a и любого эволюционного дифференцирования D_t . А с нечетными номерами все в порядке, это нетривиальные плотности, вариационная производная от них не есть тождественный 0:

In[43]:= Table[**vard**[V[n]], {n, 1, M, 2}]

Out[43]=
$$\left\{ \frac{1}{2}, \frac{u[0]}{4}, \frac{3u[0]^2}{16} + \frac{u[2]}{16}, \frac{5u[0]^3}{32} + \frac{5u[1]^2}{64} + \frac{5}{32}u[0]u[2] + \frac{u[4]}{64}, \right.$$

$$\frac{35u[0]^4}{256} + \frac{35}{128}u[0]u[1]^2 + \frac{35}{128}u[0]^2u[2] + \frac{21u[2]^2}{256} + \frac{7}{64}u[1]u[3] + \frac{7}{128}u[0]u[4] + \frac{u[6]}{256},$$

$$\frac{63u[0]^5}{512} + \frac{315}{512}u[0]^2u[1]^2 + \frac{105}{256}u[0]^3u[2] + \frac{231}{512}u[1]^2u[2] + \frac{189}{512}u[0]u[2]^2 + \frac{63}{128}u[0]u[1]u[3] +$$

$$\frac{69u[3]^2}{1024} + \frac{63}{512}u[0]^2u[4] + \frac{57}{512}u[2]u[4] + \frac{27}{512}u[1]u[5] + \frac{9}{512}u[0]u[6] + \frac{u[8]}{1024},$$

$$\frac{231u[0]^6}{2048} + \frac{1155u[0]^3u[1]^2}{1024} + \frac{1155u[1]^4}{4096} + \frac{1155u[0]^4u[2]}{2048} + \frac{2541u[0]u[1]^2u[2]}{1024} +$$

$$\frac{2079u[0]^2u[2]^2}{2048} + \frac{671u[2]^3}{2048} + \frac{693}{512}u[0]^2u[1]u[3] + \frac{1419u[1]u[2]u[3]}{1024} + \frac{759u[0]u[3]^2}{2048} +$$

$$\frac{231u[0]^3u[4]}{1024} + \frac{825u[1]^2u[4]}{2048} + \frac{627u[0]u[2]u[4]}{1024} + \frac{253u[4]^2}{4096} + \frac{297u[0]u[1]u[5]}{1024} +$$

$$\frac{209u[3]u[5]}{2048} + \frac{99u[0]^2u[6]}{2048} + \frac{121u[2]u[6]}{2048} + \frac{11}{512}u[1]u[7] + \frac{11u[0]u[8]}{2048} + \frac{u[10]}{4096} \Big\}$$

Итак, можно сформулировать такие утверждения.

Теорема. 1) Все коэффициенты в разложении (6) являются плотностями законов сохранения для КдФ; 2) плотности с чётными номерами тривиальны; 3) плотности с нечётными номерами нетривиальны.

Доказательство 1) очень простое. Очевидно, что v это плотность закона сохранения для мКдФ. Так как преобразование Миуры выполняется тождественно по параметру $\lambda = -z^2$, то все коэффициенты V_n также являются плотностями, но так как они выражены через u_n , то это плотности уже для КдФ. Из одного-единственного, самого простого закона сохранения для одного уравнения мы вытащили целую бесконечную последовательность для другого. Кстати, для мКдФ мы тоже можем теперь получить бесконечную последовательность, делая прямую подстановку (4) в V_n , но мы это пропустим.

Доказательство 2) и 3) оставляется в качестве упражнения (в файле с задачами есть подсказки).

5 Алгоритм интегрирования по частям

Алгоритм интегрирования по частям позволяет по заданной функции f от динамических переменных найти функцию g такую, что $f = D_x(g)$ или убедиться, что такой функции не существует. Сейчас мы его опишем, запрограммируем и применим для восстановления функций s_n из (7).

Замечание. Слабым местом этого алгоритма, как и формулы $E(f) = 0$, и вообще очень многих символьных вычислений является, казалось бы, безобидная операция сравнения с 0. Всегда ли легко проверить, что какое-то выражение равно или не равно 0? Для этого мы должны иметь эффективный алгоритм приведения этого выражения к некоторой однозначной канонической форме. Для полиномов

это просто – нужно раскрыть скобки и собрать коэффициенты. Для рациональных функций уже труднее – нужно приводить к общему знаменателю. А если в выражение входят какие-то алгебраические функции, или функции, удовлетворяющие какому-то набору тождеств – задача может быть очень нетривиальной. Тем более, если в выражении десятки тысяч членов и общий объем исчисляется мегабайтами, что совсем не редкость. Это нужно иметь в виду, но сейчас у нас такие сложности не возникнут, так как мы работаем только с полиномами.

Цель алгоритма в том, чтобы найти представление заданной функции f в виде

$$f = D_x(a) + b,$$

где либо $b = 0$, либо $b = b(x, u_0)$ с $\partial_0(b) \neq 0$, либо $b = b(x, u_0, \dots, u_k)$ с $\partial_k^2(b) \neq 0$. Функция `int[f]` возвращает ответ в виде пары $\{a, b\}$. В ней мы уменьшаем порядок b , пока это возможно: если на каком-то шаге имеем $b = p(x, u_0, \dots, u_{k-1})u_k + q(x, u_0, \dots, u_{k-1})$, то k можно уменьшить, заноса под D_x интеграл от p по u_{k-1} .

In[44]:=

```
(* порядок по производным *)
ord[f_] := Module[{v = vars[f]}, If[v == {}, -1, v[[-1]][[1]]]]

int[f_] := Module[{a = 0, b = Expand[f], c, k},
  While[k = ord[b]; (* вычисляем k – порядок b *)
    And[k > 0, (* выполняем цикл, пока выполняется условие, что k>0 и *)
      c = D[b, u[k]]; (* b линейна по старшей переменной *)
      Expand[D[c, u[k]]] == 0
    ],
    c = Integrate[c, u[k - 1]]; (* интегрируем по переменной, следующей по старшинству *)
    a = Expand[a + c]; (* пересчет a и b *)
    b = Expand[b - dx[c]]
  ];
  If[ord[b] == -1, (* если в результате остаток не зависит от u, то это функция от x *)
    {Expand[a + Integrate[b, x]], 0},
    {a, b}
  ]
]
```

Возьмем какое-нибудь выражение и посчитаем от него вариационную производную.

In[46]:=

```
f = u[0] u[5] u[3]
varD[f]
```

Out[46]= $u[0] u[3] u[5]$ Out[47]= $-5 u[4]^2 - 11 u[3] u[5] - 13 u[2] u[6] - 8 u[1] u[7] - 2 u[0] u[8]$

Получилось не 0, значит f нельзя представить в виде полной производной. Применим интегрирование по частям. Получится пара, в которой второй элемент это то, что не лезет под знак D_x , из-за нелинейности по старшей букве.

In[48]:=

```
ab = int[f]
Expand[f - dx[ab[[1]]] - ab[[2]]]
```

Out[48]= $\{u[0] u[3] u[4], -u[1] u[3] u[4] - u[0] u[4]^2\}$

Out[49]= 0

А в этих примерах проинтегрировать получается.

```
In[50]:= f = g[u[0]] u[1]
          var[d[f]]
          int[f]
```

```
Out[50]= g[u[0]] u[1]
```

```
Out[51]= 0
```

```
Out[52]= {∫ g[u[0]] du[0], 0}
```

```
In[53]:= f = dx[x^3 u[0] u[1] + u[2] u[7] u[4]]
          var[d[f]]
          int[f]
```

```
Out[53]= 3 x^2 u[0] u[1] + x^3 u[1]^2 + x^3 u[0] u[2] + u[3] u[4] u[7] + u[2] u[5] u[7] + u[2] u[4] u[8]
```

```
Out[54]= 0
```

```
Out[55]= {x^3 u[0] u[1] + u[2] u[4] u[7], 0}
```

Проверяем V_n с чётными номерами. Да, действительно, в ответах вторая компонента равна 0, то есть всё занеслось под D_x :

```
In[56]:= Table[int[2^n V[n]] /. u[k_] -> u_k, {n, 2, M, 2}]
```

```
Out[56]= {{u_0, 0}, {2 u_0^2 + u_2, 0}, {16 u_0^3 / 3 + 5 u_1^2 + 8 u_0 u_2 + u_4, 0},
          {16 u_0^4 + 60 u_0 u_1^2 + 48 u_0^2 u_2 + 20 u_2^2 + 28 u_1 u_3 + 12 u_0 u_4 + u_6, 0}, {256 u_0^5 / 5 + 480 u_0^2 u_1^2 + 256 u_0^3 u_2 +
          452 u_1^2 u_2 + 320 u_0 u_2^2 + 448 u_0 u_1 u_3 + 69 u_3^2 + 96 u_0^2 u_4 + 112 u_2 u_4 + 54 u_1 u_5 + 16 u_0 u_6 + u_8, 0},
          {512 u_0^6 / 3 + 3200 u_0^3 u_1^2 + 1130 u_1^4 + 1280 u_0^4 u_2 + 9040 u_0 u_1^2 u_2 + 3200 u_0^2 u_2^2 + 3904 u_2^3 / 3 +
          4480 u_0^2 u_1 u_3 + 5620 u_1 u_2 u_3 + 1380 u_0 u_3^2 + 640 u_0^3 u_4 + 1640 u_1^2 u_4 + 2240 u_0 u_2 u_4 +
          252 u_4^2 + 1080 u_0 u_1 u_5 + 418 u_3 u_5 + 160 u_0^2 u_6 + 240 u_2 u_6 + 88 u_1 u_7 + 20 u_0 u_8 + u_10, 0}}
```

Проверяем V_n с нечётными номерами. Дифференцируем по t , в силу КдФ, и заносим под D_x , опять вторая компонента равна 0, а первая это и есть искомые функции s_n :

In[57]:=

Table[int[star[2ⁿ V[n], u[3] + 6 u[0] u[1]]] /. u[k_] => u_k, {n, 2, M, 2}]

Out[57]= $\left\{ \left\{ 6 u_0 u_1 + u_3, 0 \right\}, \left\{ 24 u_0^2 u_1 + 18 u_1 u_2 + 10 u_0 u_3 + u_5, 0 \right\}, \right.$
 $\left\{ 96 u_0^3 u_1 + 60 u_1^3 + 252 u_0 u_1 u_2 + 64 u_0^2 u_3 + 68 u_2 u_3 + 40 u_1 u_4 + 14 u_0 u_5 + u_7, 0 \right\},$
 $\left\{ 384 u_0^4 u_1 + 1080 u_0 u_1^3 + 2160 u_0^2 u_1 u_2 + 1224 u_1 u_2^2 + 352 u_0^3 u_3 + 900 u_1^2 u_3 + \right.$
 $1224 u_0 u_2 u_3 + 720 u_0 u_1 u_4 + 250 u_3 u_4 + 120 u_0^2 u_5 + 166 u_2 u_5 + 70 u_1 u_6 + 18 u_0 u_7 + u_9, 0 \left. \right\},$
 $\left\{ 1536 u_0^5 u_1 + 11520 u_0^2 u_1^3 + 14976 u_0^3 u_1 u_2 + 13560 u_1^3 u_2 + 26928 u_0 u_1 u_2^2 + 1792 u_0^4 u_3 + \right.$
 $19800 u_0 u_1^2 u_3 + 13056 u_0^2 u_2 u_3 + 9524 u_2^2 u_3 + 7000 u_1 u_3^2 + 7680 u_0^2 u_1 u_4 +$
 $11140 u_1 u_2 u_4 + 5500 u_0 u_3 u_4 + 832 u_0^3 u_5 + 2720 u_1^2 u_5 + 3652 u_0 u_2 u_5 + 922 u_4 u_5 +$
 $1540 u_0 u_1 u_6 + 658 u_3 u_6 + 192 u_0^2 u_7 + 328 u_2 u_7 + 108 u_1 u_8 + 22 u_0 u_9 + u_{11}, 0 \left. \right\},$
 $\left\{ 6144 u_0^6 u_1 + 96000 u_0^3 u_1^3 + 27120 u_1^5 + 92160 u_0^4 u_1 u_2 + 352560 u_0 u_1^3 u_2 + 342720 u_0^2 u_1 u_2^2 + 171432 u_1 u_2^3 + \right.$
 $8704 u_0^5 u_3 + 252000 u_0^2 u_1^2 u_3 + 108800 u_0^3 u_2 u_3 + 377200 u_1^2 u_2 u_3 + 247624 u_0 u_2^2 u_3 + 182000 u_0 u_1 u_3^2 +$
 $26460 u_3^3 + 64000 u_0^3 u_1 u_4 + 73400 u_1^3 u_4 + 289640 u_0 u_1 u_2 u_4 + 70000 u_0^2 u_3 u_4 + 126120 u_2 u_3 u_4 +$
 $36880 u_1 u_4^2 + 5120 u_0^4 u_5 + 70720 u_0 u_1^2 u_5 + 46480 u_0^2 u_2 u_5 + 41668 u_2^2 u_5 + 61348 u_1 u_3 u_5 + 23972 u_0 u_4 u_5 +$
 $19600 u_0^2 u_1 u_6 + 34804 u_1 u_2 u_6 + 17108 u_0 u_3 u_6 + 3430 u_5 u_6 + 1600 u_0^3 u_7 + 6392 u_1^2 u_7 + 8528 u_0 u_2 u_7 +$
 $2572 u_4 u_7 + 2808 u_0 u_1 u_8 + 1428 u_3 u_8 + 280 u_0^2 u_9 + 570 u_2 u_9 + 154 u_1 u_{10} + 26 u_0 u_{11} + u_{13}, 0 \left. \right\} \right\}$