В.Э. Адлер · Классические интегрируемые системы

Лекция 10 · 10 апреля 2023

Некоторые вычисления на *Mathematica*

1 Построение графиков

1.1 Plot. Гармонический осциллятор

Определяем собственную функцию основного состояния и последующие при помощи оператора рождения.

- присваивание
- определение функции
- Factor
- дифференцирование D
- тест “такой, что”

In[1]:=
```
F[0] = Exp[-x^2/2];
F[n_] := Factor[-D[F[n - 1], x] + x F[n - 1]] /; n > 0
```

Можно убедиться, что результат выражается через многочлены Эрмита (они встроены в *Mathematica*, как и многие другие спецфункции).

- Table – создание списка
- % – предыдущий результат
- применение Expand “сзади”

In[3]:=
```
F[0]
F[1]
F[2]
F[3]
F[4]
Table[F[j] - HermiteH[j, x] F[0], {j, 0, 5}]
% // Expand
```

Out[3]= $e^{-\frac{x^2}{2}}$

Out[4]= $2\, e^{-\frac{x^2}{2}}\, x$

Out[5]= $2\, e^{-\frac{x^2}{2}}\, \left(-1 + 2\, x^2\right)$

Out[6]= $4\, e^{-\frac{x^2}{2}}\, x\, \left(-3 + 2\, x^2\right)$

Out[7]= $4\, e^{-\frac{x^2}{2}}\, \left(3 - 12\, x^2 + 4\, x^4\right)$

Out[8]= $\left\{0,\ 0,\ 2\, e^{-\frac{x^2}{2}}\, \left(-1 + 2\, x^2\right) - e^{-\frac{x^2}{2}}\, \left(-2 + 4\, x^2\right),\ 4\, e^{-\frac{x^2}{2}}\, x\, \left(-3 + 2\, x^2\right) - e^{-\frac{x^2}{2}}\, \left(-12\, x + 8\, x^3\right),\right.$
$\left. 4\, e^{-\frac{x^2}{2}}\, \left(3 - 12\, x^2 + 4\, x^4\right) - e^{-\frac{x^2}{2}}\, \left(12 - 48\, x^2 + 16\, x^4\right),\ 8\, e^{-\frac{x^2}{2}}\, x\, \left(15 - 20\, x^2 + 4\, x^4\right) - e^{-\frac{x^2}{2}}\, \left(120\, x - 160\, x^3 + 32\, x^5\right)\right\}$

Out[9]= $\{0, 0, 0, 0, 0, 0\}$

Определяем нормировочные константы.

- Интегрирование

In[10]:=
```
c[n_] := Integrate[F[n]^2, {x, -∞, ∞}]^(1/2)
Table[c[j], {j, 0, 5}]
```

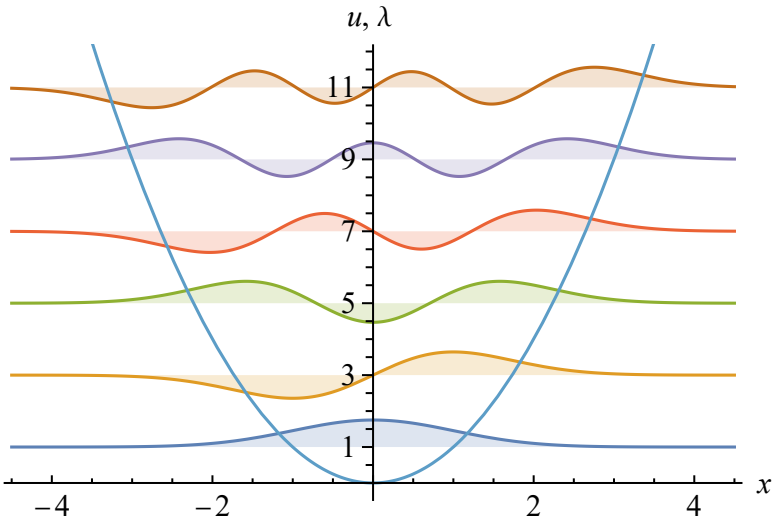
Out[11]=
$\left\{\pi^{1/4},\ \sqrt{2}\, \pi^{1/4},\ 2\, \sqrt{2}\, \pi^{1/4},\ 4\, \sqrt{3}\, \pi^{1/4},\ 8\, \sqrt{6}\, \pi^{1/4},\ 16\, \sqrt{15}\, \pi^{1/4}\right\}$

Строим график.

- Plot и некоторые опции

In[12]:=
```
osc = Plot[ (* список выводимых функций *)
  Evaluate[Append[Table[F[j] / c[j] + 2 j + 1, {j, 0, 5}], x^2]],
  (* переменная *)
  {x, -4.5, 4.5},
  (* опции *)
  Filling → Table[j → 2 j - 1, {j, 0, 6}], (* заливка *)
  Ticks → {Automatic, (* метки по оси x *)
    Table[If[OddQ[i], {i, i, {0, .015}}, {i, , {0, .01}}], {i, 0, 13, 1/2}]},
  (* метки по оси y *)
  PlotRange → {{-4.6, 4.6}, {-0.5, 12.2}}, (* выводимый прямоугольник *)
  ImageSize → 400, (* выводимый прямоугольник *)
  AxesLabel → {"x", "u, λ"}, (* метки на осях *)
  BaseStyle → {FontSize → 16, FontFamily → "Times New Roman"} ] (* шрифт *)
```

Out[12]=



Запись изображения на диск.

- <> конкатенация строк

In[ ]:=
```
Export[NotebookDirectory[] <> "osc.pdf", osc, "PDF"];
```

1.2 ListPlot и Manipulate. Солитон цепочки Вольтерры

Определим функцию $u_n(t)$ по явным формулам и проверим, что она удовлетворяет цепочке $u_{n,t} = u_n(u_{n+1} - u_{n-1})$.

- Clear – очистка определений
- Together – приведение к общему знаменателю

In[13]:=
```
Clear[a, c, t]
w[n_] := 1 + c a^n Exp[ (a - a^-1) t];
u[n_] := w[n + 1] × w[n - 2] / w[n] × w[n - 1]
Together[D[u[n], t] - u[n] (u[n + 1] - u[n - 1])]
```

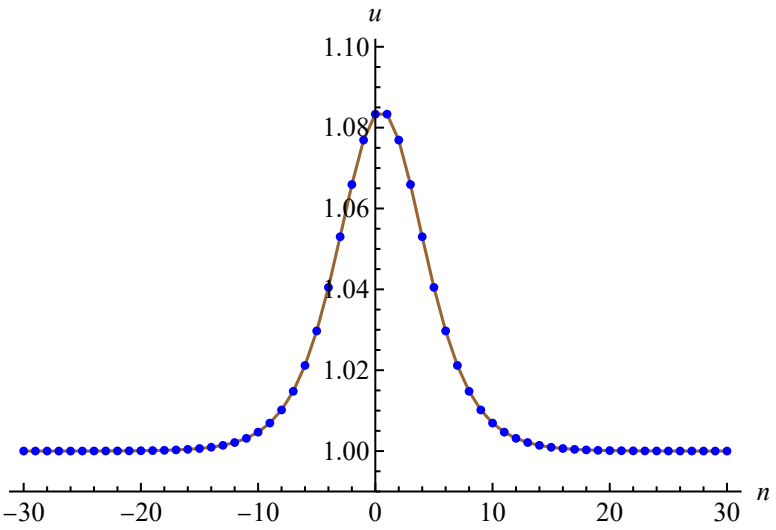
Out[16]=
```
0
```

Определим функцию plot. Она выводит график при фиксированных значениях a, c, t. Для наглядности, два раза – в виде точек и в виде линии.

■ Пример лямбда-функции: {#, #} &[x] даёт {x, x}

In[17]:=
```
pl[a0_, c0_, t0_] :=
 ListPlot[
  (* список точек (два раза) *)
  {#, #} &[Table[{n, u[n] /. {t → t0, a → a0, c → c0}}, {n, -30, 30}]],
  (* опции *)
  PlotRange → {0.99, 1.102},
  Joined → {True, False},  (* точки из первого списка соединяем, из второго нет *)
  PlotStyle → {Brown, {Blue, PointSize[0.012]}}, (* стиль, для каждого списка *)
  BaseStyle → {FontSize → 14, FontFamily → "Times New Roman"},
  AxesLabel → {"n", "u"},
  ImageSize → 400]

pl[1.5, 1, 0]
```

Out[18]=

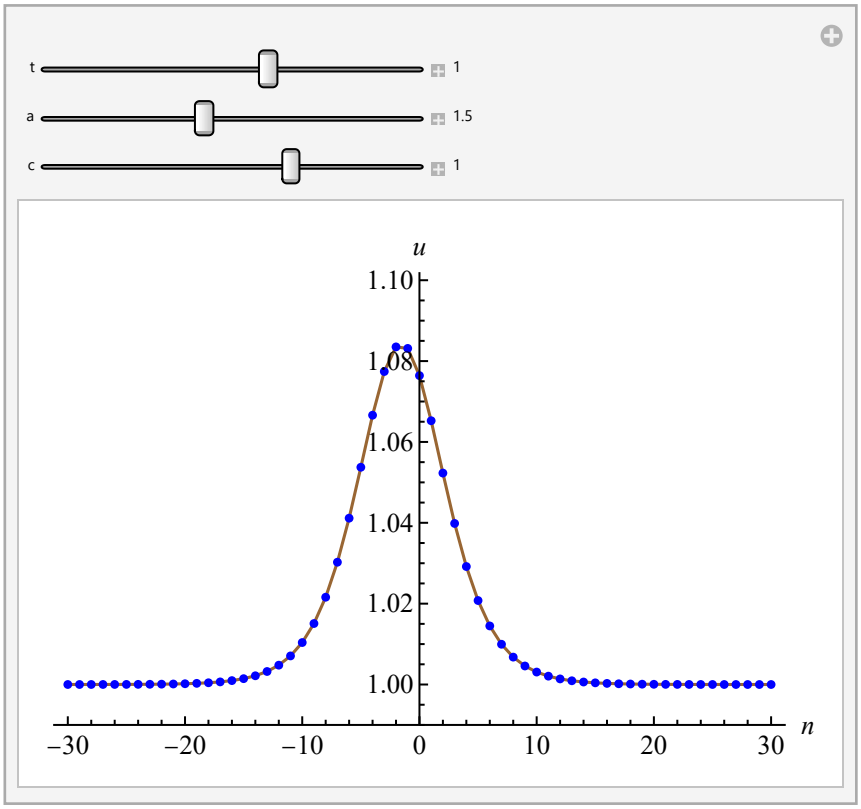


Запустим график в манипуляторе.

■ Динамический контент желательно удалять после использования.

In[19]:=
```
Manipulate[pl[a, c, t],
 {{t, 1}, -5, 5, Appearance → "Labeled"},
 {{a, 1.5}, -1, 5, Appearance → "Labeled"},
 {{c, 1}, -3, 3, Appearance → "Labeled"}]
```

Out[19]=



1.3 Plot3D. Солитон огибающей (NLS$^+$)

Комплексные числа

- i (а также π, e) – зарезервированные обозначения
- ComplexExpand предполагает, что все буквы обозначат вещественные переменные

In[20]:=
```
Clear[a, b, c, d]

u = a Exp[i (b x + (b^2 - a^2) t + c)] / Cosh[a (x + 2 b t) + d]
v = u /. i → -i

Simplify[-i D[u, t] + D[u, x, x] + 2 u^2 v]  (* проверили, что это решение НУШ *)

Simplify[ComplexExpand[ReIm[u]]] (* вещественная и мнимая части решения *)
```

Out[21]=
$$a\, e^{i \left(c+\left(-a^2+b^2\right) t+b x\right)} \text{Sech}[d+a (2 b t+x)]$$

Out[22]=
$$a\, e^{-i \left(c+\left(-a^2+b^2\right) t+b x\right)} \text{Sech}[d+a (2 b t+x)]$$

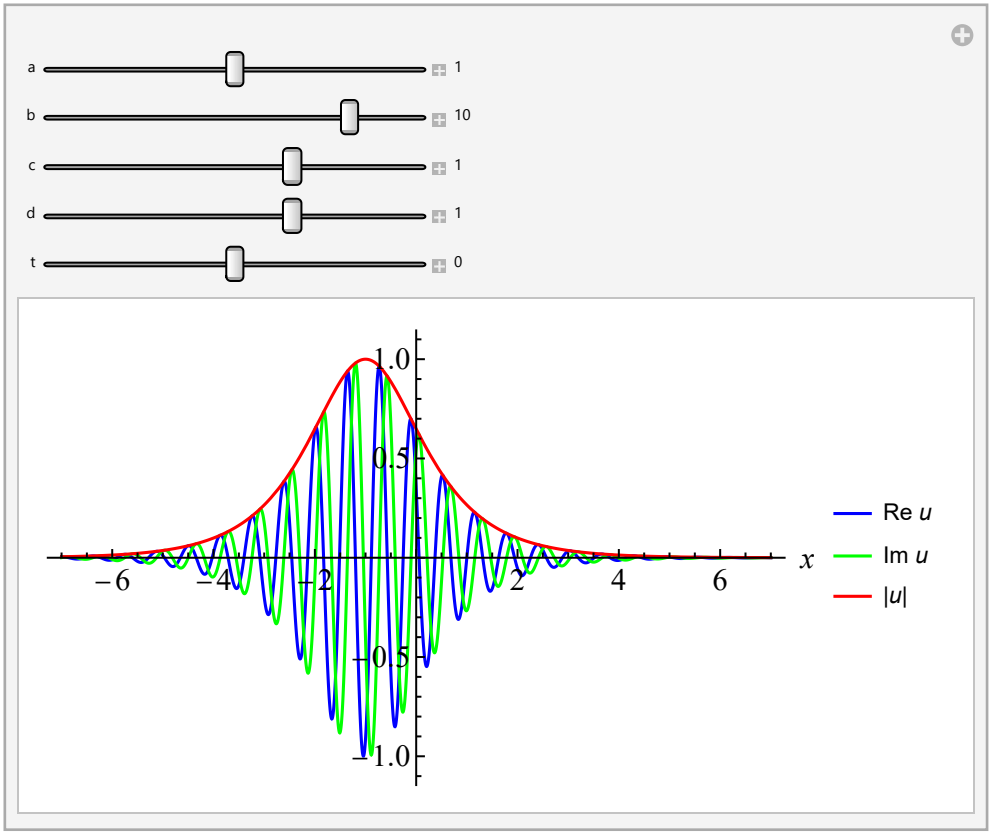
Out[23]=
$$0$$

Out[24]=
$$\left\{a \text{Cos}\left[c-a^2 t+b (b t+x)\right] \text{Sech}[d+a (2 b t+x)],\ a \text{Sech}[d+a (2 b t+x)] \text{Sin}\left[c-a^2 t+b (b t+x)\right]\right\}$$

In[25]:=

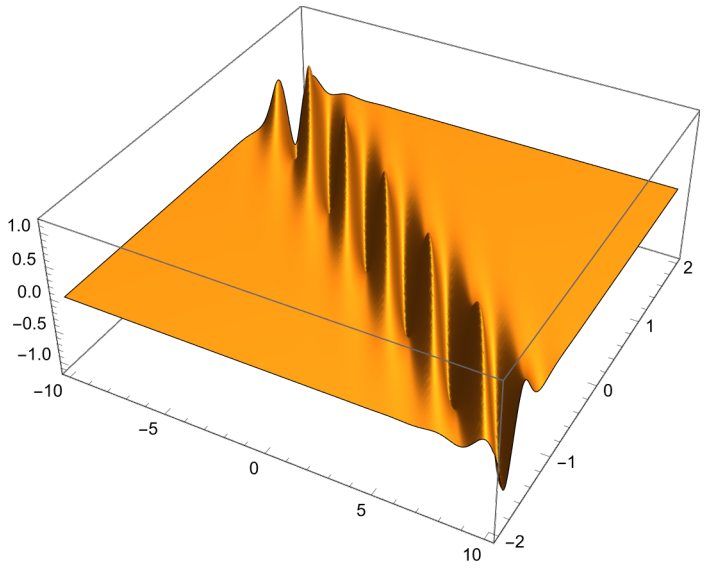
```
plot[a_, b_, c_, d_, t_] :=
 Module[{p = a Cos[b x + (b^2 - a^2) t + c] / Cosh[a (x + 2 b t) + d],
   q = a Sin[b x + (b^2 - a^2) t + c] / Cosh[a (x + 2 b t) + d]},
  Plot[{p, q, Sqrt[p^2 + q^2]}, {x, -7, 7},
   PlotStyle → {Blue, Green, Red},
   PlotRange → {-1.15, 1.15},
   BaseStyle → {FontSize → 16, FontFamily → "Times New Roman"},
   ImageSize → 400,
   AxesLabel → {"x", None},
   PlotLegends → LineLegend[{Blue, Green, Red}, {"Re u", "Im u", "|u|"}]
  ]]

Manipulate[
 plot[a, b, c, d, t],
 {{a, 1}, 0, 2, Appearance → "Labeled"},
 {{b, 10}, -15, 15, Appearance → "Labeled"},
 {{c, 1}, -3, 3, Appearance → "Labeled"},
 {{d, 1}, -3, 3, Appearance → "Labeled"},
 {{t, 0}, -1, 1, Appearance → "Labeled"}]
```

Out[26]=



In[27]:=
```
Plot3D[a Cos[b x + (b^2 - a^2) t + c] / Cosh[a (x + 2 b t) + d] /. {a → 1, b → 3, c → 0, d → 0},
 {x, -10, 10}, {t, -2, 2},
 PlotPoints → 100,
 Mesh → False,
 PlotRange → {-1.2, 1.2}]
```

Out[27]=



1.4 ParametricPlot, ParametricPlot3D. Уравнение Хопфа

$$u_t = u\, u_x \quad \Leftrightarrow \quad u(x, t) = f(x + u(x, t)\, t)$$

Проверим, что это действительно решение.

■ В данном случае переменные с индексом – просто обозначение, оно никак не интерпретируется машиной. Можно было бы обозначить их вместо u_x. Использование индексов иногда приводит к недоразумениям.

In[28]:=
```
Clear[x, y, t, u, f]
eq = u - f[x + u t];
eqx = D[eq, x] + D[eq, u] u_x
eqt = D[eq, t] + D[eq, u] u_t
Factor[eqt - u eqx]

Subscript[u, x]
D[%, x]
```

Out[30]=
$$-f'[t\,u + x] + u_x\,(1 - t\,f'[t\,u + x])$$

Out[31]=
$$-u\,f'[t\,u + x] + u_t\,(1 - t\,f'[t\,u + x])$$

Out[32]=
$$(-u_t + u\,u_x)\,(-1 + t\,f'[t\,u + x])$$

Out[33]=
$$u_x$$

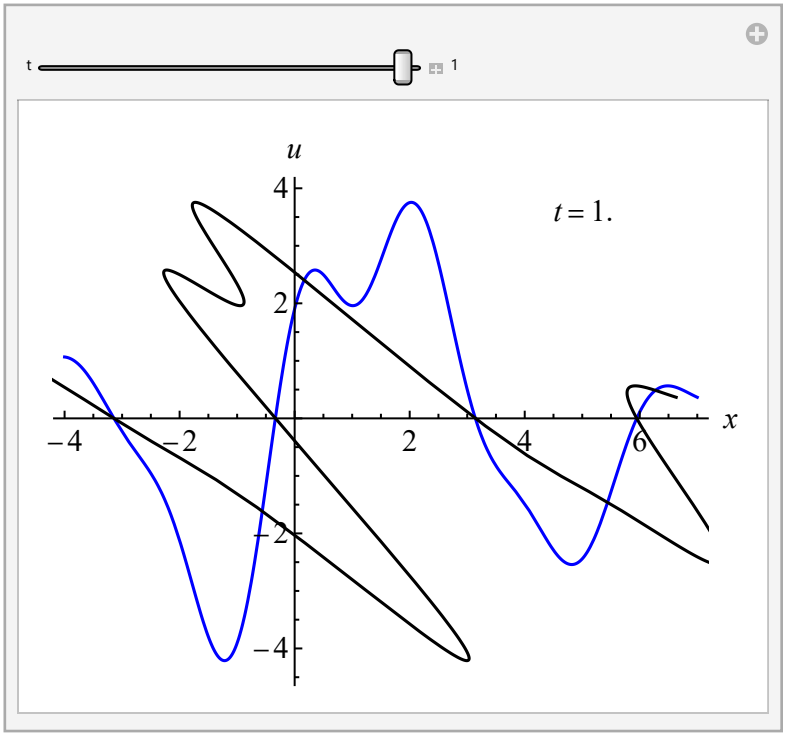
Out[34]=
$$\text{Subscript}^{(0,1)}[u, x]$$

Зададим начальное условие $u(x, 0) = f(x)$. Построим кривые $(x(s), u(s))$, где параметр $s = x + u\,t$. Так как $u = f(s)$, то $x = s - t\,f(s)$.

■ N[expr] переводит все точные числа в выражении в формат float
ToString[expr] возвращает строку из числа (или выражения)

In[35]:=
```
f[x_] := Exp[-0.05 (x - 1)^2] (4 Sin[x] + Cos[2 x] + Cos[3 x])

g1[t_] := ParametricPlot[
  (* две кривые: при t=0 и при заданном t *)
  {{s, f[s]}, {s - f[s] t, f[s]}},
  (* интервал изменения параметра, совпадает с интервалом для x *)
  {s, -4, 7},
  (* опции *)
  PlotStyle → {Blue, Black},
  PlotRange → {{-4.2, 7.2}, Automatic},
  AspectRatio → Automatic, (* одинаковый масштаб по осям *)
  AxesLabel → {"x", "u"},
  Epilog → {Text["t = " <> ToString[N[t]], {4.5, 3.6}, {-1, 0}]}, (* вывод текста *)
  BaseStyle → {FontSize → 16, FontFamily → "Times New Roman"}]

Manipulate[g1[t], {{t, 1}, -1, 1, Appearance → "Labeled"}]
```

Out[37]=



То же самое можно сделать в 3D.

$$u_t = u\,u_x + u^2\,u_y \qquad u(x, y, t) = f\left(x + u(x, y, t)\,t, y + u^2(x, y, t)\,t\right)$$

Проверка

In[38]:=
```
Clear[f]
eq = u - f[x + u t, y + u^2 t];
eqx = D[eq, x] + D[eq, u] u_x
eqy = D[eq, y] + D[eq, u] u_y
eqt = D[eq, t] + D[eq, u] u_t
Factor[eqt - u eqx - u^2 eqy]
```

Out[40]=
$$-f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]+u_x\left(1-2\,t\,u\,f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]-t\,f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]\right)$$

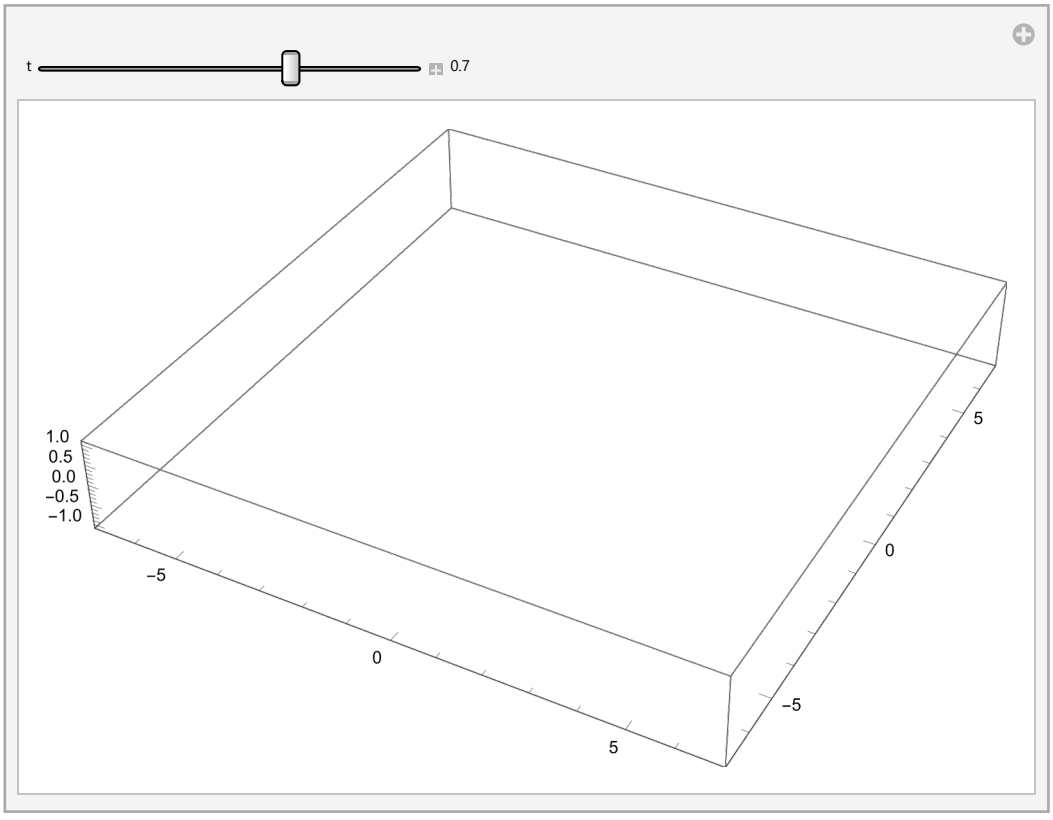
Out[41]=
$$-f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]+u_y\left(1-2\,t\,u\,f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]-t\,f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]\right)$$

Out[42]=
$$-u^2\,f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]-u\,f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]+ u_t\left(1-2\,t\,u\,f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]-t\,f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]\right)$$

Out[43]=
$$\left(-u_t+u\,u_x+u^2\,u_y\right)\left(-1+2\,t\,u\,f^{(0,1)}\left[t\,u+x,\ t\,u^2+y\right]+t\,f^{(1,0)}\left[t\,u+x,\ t\,u^2+y\right]\right)$$

In[44]:=
```
f[x_, y_] := Sin[x] Cos[y]

Manipulate[
 ParametricPlot3D[{sx - f[sx, sy] t, sy - f[sx, sy]^2 t, f[sx, sy]}, {sx, -7, 7}, {sy, -7, 7},
  PlotRange → {{-7, 7}, {-7, 7}, {-1.1, 1.1}},
  AspectRatio → Automatic,
  ImageSize → 500],
 {{t, 0.7}, -2, 2, Appearance → "Labeled"}]
```

Out[45]=

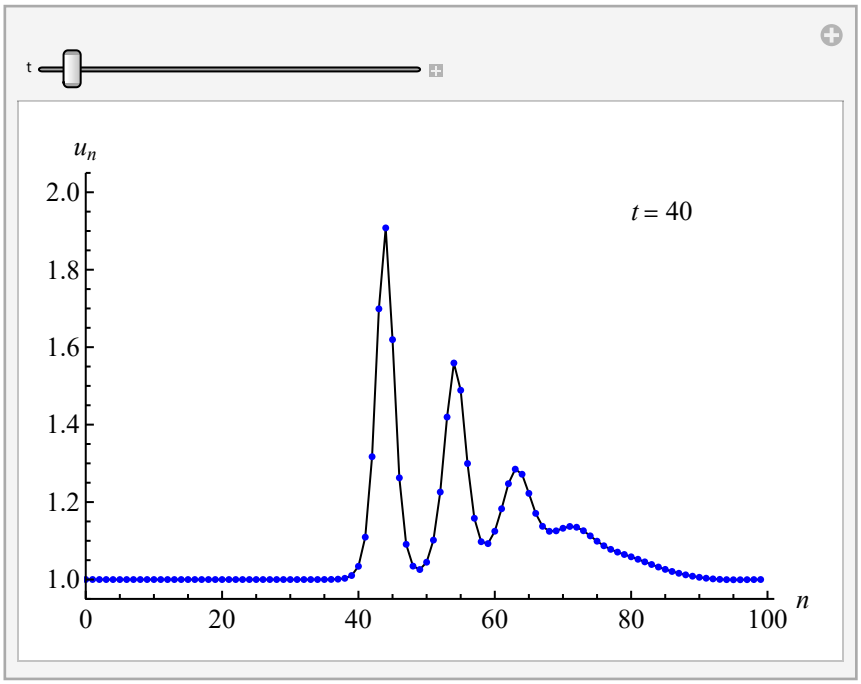


2 Численное решение ОДУ

2.1 NDSolve, ListPlot. Цепочка Вольтерры

In[46]:=
```
(* задаём период и время счёта *)
M1 = 100;
t1 = 1000;

(* уравнения, с учётом периодического замыкания с периодом M *)
eqs = Table[u[n]'[t] == u[n][t] (u[Mod[n + 1, M1]][t] - u[Mod[n - 1, M1]][t]), {n, 0, M1 - 1}];

(* задаём начальные условия. Попробуйте что-нибудь поменять *)
ic = Table[{u[n][0] == 1 + 0.5 Exp[-0.01 (n - M1 / 2)^2]}, {n, 0, M1 - 1}];

(* строим решение при помощи NDSolve *)
sol1 = NDSolve[Join[eqs, ic], Table[u[n], {n, 0, M1 - 1}], {t, 0, t1}, MaxSteps → ∞][[1]];

(* массив точек *)
u1[t_] := Table[{n, u[n][t]}, {n, 0, M1 - 1}] /. sol1

(* функция для вывода. Для наглядности, массив выводится два раза: как ломаная линия
   и как набор жирных точек в ее вершинах *)
plotv1[t_] := ListPlot[{u1[t], u1[t]},
  PlotRange → {{-0.1, M1 + 1}, {0.95, 2.05}},
  PlotStyle → {{Thin, Black}, {Blue, PointSize[0.01]}},
  Joined → {True, False},
  AxesLabel → {" n ", "u_n"},
  BaseStyle → {FontSize → 14, FontFamily → "Times New Roman"},
  ImageSize → 400,
  Epilog → {Text["t = " <> ToString[t], {0.8 M1, 1.95}, {-1, 0}]}
 ]

(* анимация *)
Manipulate[plotv1[t], {{t, 40}, 0, t1}]
```

Out[53]=



Запись кадров в пдф-файлы, для последующей вставки в латех: (это довольно долго)

```
Do[
 Export[NotebookDirectory[] <> "VL_" <> ToString[j] <> ".pdf", plotv1[j], "PDF"], {j, 0, 100}]
```

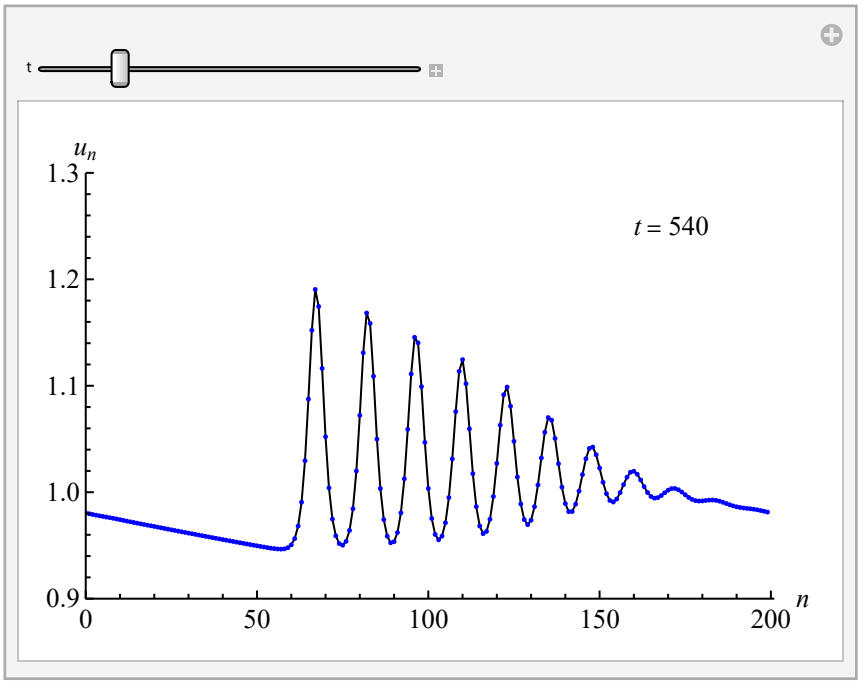
Также можно сохранить список кадров в виде гиф-файла:

In[ ]:=
```
Export[NotebookDirectory[] <> "VL.gif", Table[plotv1[j], {j, 0, 100}], "GIF"];
```

Если взять начальное условие с более широким горбиком, то мы придем примерно к той картине, что была в эксперименте Забуски-Краскала для КдФ. Напомним, что там решалось чисто разностное уравнение с начальными условиями в виде синусоидальной волны с большим периодом (порядка 100 узлов решетки) и не очень большой амплитудой. Такое начальное условие распадалось на несколько солитонов плюс “рябь” (число солитонов зависит от начальной амплитуды), которые двигались с разной скоростью и, благодаря периодичности, всё время нагоняли друг друга. Качественно, в цепочке Вольтерры наблюдается точно такая же картина.

In[54]:=
```
M2 = 200;
t2 = 3000;

eqs = Table[u[n]'[t] == u[n][t] (u[Mod[n + 1, M2]][t] - u[Mod[n - 1, M2]][t]), {n, 0, M2 - 1}];
ic = Table[{u[n][0] == 1 + 0.07 Cos[2 π / M2 n]}, {n, 0, M2 - 1}];
sol2 = NDSolve[Join[eqs, ic], Table[u[n], {n, 0, M2 - 1}], {t, 0, t2}, MaxSteps → ∞][[1]];

u2[t_] := Table[{n, u[n][t]}, {n, 0, M2 - 1}] /. sol2

plotv2[t_] := ListPlot[{u2[t], u2[t]},
  PlotRange → {{-0.1, M2 + 1}, {0.9, 1.3}},
  PlotStyle → {{Thin, Black}, {Blue, PointSize[0.007]}},
  Joined → {True, False},
  AxesLabel → {" n", "u_n"},
  BaseStyle → {FontSize → 14, FontFamily → "Times New Roman"},
  ImageSize → 400,
  Epilog → {Text["t = " <> ToString[t], {0.8 M2, 1.25}, {-1, 0}]}
 ]

Manipulate[plotv2[t], {{t, 540}, 0, t2}]
```

Out[61]=



2.2 NDSolve, ListPlot3D. Коммутирующие векторные поля, 2-зонное решение КдФ

Задача. Пусть $u_n = \partial_x^n(u)$. Рассмотрим уравнение Новикова

(1) $u_4 - 10\,u\,u_2 - 5\,u_1^2 + 10\,u^3 + c_1\left(u_2 - 3\,u^2\right) + c_2\,u + c_3 = 0,$

где c_1, c_2, c_3 произвольные постоянные (это один раз проинтегрированное стационарное уравнение для симметрии КдФ пятого порядка плюс младшие симметрии). Проверить,

что (1) совместно с уравнением КдФ

(2) $u_t = u_3 - 6\,u\,u_1$.

Построить график любого решения $u(x, t)$ удовлетворяющего обоим уравнениям.

Решение.

Первый способ символьной проверки. Определим эволюционное дифференцирование D_t на бесконечном наборе u_n и проверим выполнение тождества

$$D_t(G) = \left(D_x^3 - 6\,u\,D_x\right)(G),$$

где G – левая часть (1). Тогда, если $G = 0$, то и $D_t(G) = 0$, что и требуется.

■ Block – объявляем переменную a локальной.

In[62]:=
```
dif[f_] := Block[{a}, D[f /. u[n_] :> u[n] + a du[n], a] /. a → 0]
dx[f_] := dif[f] /. du[n_] :> u[n + 1]
dx[f_, n_] := Nest[dx, f, n]

ut = u[3] - 6 u[0] × u[1];
dt[f_] := dif[f] /. du[n_] :> dx[ut, n]

G = u[4] - 10 u[0] × u[2] - 5 u[1]^2 + 10 u[0]^3 + c1 (u[2] - 3 u[0]^2) + c2 u[0] + c3;
Expand[dt[G] - dx[G, 3] + 6 u[0] × dx[G]]
```

Out[68]=
```
0
```

Второй способ символьной проверки. Перепишем оба уравнения в виде конечномерных динамических систем. Так как четвертая производная выражается из ОДУ (1) через младшие, то динамическими переменными являются $u = u_0, u_1, u_2, u_3$. Сразу можно написать векторное поле D_x (точнее – то, как оно действует на произвольную функцию от динамических переменных).

После этого определяем D_t. Уравнение (1) задаёт правило дифференцирования по t только для переменной u_0, а нам нужно ещё для u_1, u_2, u_3. Значит, нужно *продолжить* дифференцирование D_t на эти производные, пользуясь тем, что D_x уже определено и полагая, по определению, что $D_t(u_k) = D_x^k(u_t)$. Так мы и делали выше при определении D_t. Сейчас лишь заменяем D_x на укороченное векторное поле, действующее на четырех переменных.

In[69]:=
```
dx[f_] := u[1] × D[f, u[0]] + u[2] × D[f, u[1]] + u[3] × D[f, u[2]] +
   (10 u[0] × u[2] + 5 u[1]^2 - 10 u[0]^3 - c1 (u[2] - 3 u[0]^2) - c2 u[0] - c3) D[f, u[3]]

dt[f_] := ut D[f, u[0]] + dx[ut] × D[f, u[1]] + dx[ut, 2] × D[f, u[2]] + dx[ut, 3] × D[f, u[3]]
```

Можно посмотреть на компоненты векторного поля D_t:

In[71]:=
```
dt[u[0]]
Collect[dt[u[1]], {u[3], u[2], u[1], u[0]}, Factor]
Collect[dt[u[2]], {u[3], u[2], u[1], u[0]}, Factor]
Collect[dt[u[3]], {u[3], u[2], u[1], u[0]}, Factor]
```

Out[71]=
$$-6\,u[0]\times u[1]+u[3]$$

Out[72]=
$$-c3-c2\,u[0]+3\,c1\,u[0]^2-10\,u[0]^3-u[1]^2+(-c1+4\,u[0])\,u[2]$$

Out[73]=
$$\left(-c2+6\,c1\,u[0]-30\,u[0]^2\right)u[1]+2\,u[1]\times u[2]+(-c1+4\,u[0])\,u[3]$$

Out[74]=
$$c1\,c3+(c1\,c2-4\,c3)\,u[0]+\left(-3\,c1^2-4\,c2\right)u[0]^2+22\,c1\,u[0]^3-40\,u[0]^4+$$
$$(c1-40\,u[0])\,u[1]^2+\left(c1^2-c2-8\,c1\,u[0]+10\,u[0]^2\right)u[2]+2\,u[2]^2+6\,u[1]\times u[3]$$

Итак, построили два векторных поля. Совместность означает, что они коммутируют. Считаем коммутатор. Для этого нужно вычислить его значения на координатных функциях (= динамических переменных), то есть, $[D_x, D_t]\,(u_k),\ k = 0, 1, 2, 3$:

In[75]:=
```
Table[Expand[dx[dt[u[k]]] - dt[dx[u[k]]]], {k, 0, 3}]
```

Out[75]=
$$\{0, 0, 0, 0\}$$

Всё сошлось. На самом деле, ясно, что равенство нулю коммутатора на первых трёх координатах выполняется автоматически – в силу построения дифференцирования D_t по формуле $D_t(u_k) = D_x^k(u_t)$. Для примера, испортим что-то в уравнениях. Тогда коммутативность нарушится, но только на последней координате. Так что, можно было бы проверять только её.

In[76]:=
```
(* портим ut *)
ut = u[3] - 66 u[0]×u[1];
Table[Expand[dx[dt[u[k]]] - dt[dx[u[k]]]], {k, 0, 3}]

(* откатываем назад *)
ut = u[3] - 6 u[0]×u[1];
Table[Expand[dx[dt[u[k]]] - dt[dx[u[k]]]], {k, 0, 3}]
```

Out[77]=
$$\{0, 0, 0, 300\,c3\,u[1]+300\,c2\,u[0]\times u[1]-900\,c1\,u[0]^2\,u[1]+3000\,u[0]^3\,u[1]-$$
$$900\,u[1]^3+120\,c1\,u[1]\times u[2]-1200\,u[0]\times u[1]\times u[2]-600\,u[2]\times u[3]\}$$

Out[79]=
$$\{0, 0, 0, 0\}$$

Численное построение решения. (Более эффективный счёт получается при переходе к уравнениям Дубровина, которые дают также аналитический метод решения, см. лекцию 5. Но, здесь мы этого не касаемся.) Определяем решение задачи Коши по x:

In[80]:=
```
nsolx[{u00_, u10_, u20_, u30_}, {c1_, c2_, c3_}, {x0_, x1_}] :=
 NDSolve[
   {u[0]'[x] == u[1][x],
    u[1]'[x] == u[2][x],
    u[2]'[x] == u[3][x],
    u[3]'[x] == 10 u[0][x] u[2][x] +
      5 u[1][x]^2 - 10 u[0][x]^3 - c1 (u[2][x] - 3 u[0][x]^2) - c2 u[0][x] - c3,
    u[0][x0] == u00,
    u[1][x0] == u10,
    u[2][x0] == u20,
    u[3][x0] == u30},
   {u[0], u[1], u[2], u[3]},
   {x, x0, x1}][[1]]
```

Определяем решение задачи Коши по *t*. Нам нужно добавить аргумент *t* к тем обозначениям, что у нас были. Сделаем это в следующей ячейке, а потом скопируем в аргументы NDSolve. (Конечно, такое рукоделие не есть хорошо. Для более серьезных задач нужно делать по другому, чтобы все было автоматически.)

In[81]:=
```
dt[u[0]] /. u[k_] :> u[k][t]
Expand[dt[u[1]]] /. u[k_] :> u[k][t]
Expand[dt[u[2]]] /. u[k_] :> u[k][t]
Expand[dt[u[3]]] /. u[k_] :> u[k][t]
```

Out[81]=
$$-6\,u[0][t]\times u[1][t] + u[3][t]$$

Out[82]=
$$-c3 - c2\,u[0][t] + 3\,c1\,u[0][t]^2 - 10\,u[0][t]^3 - u[1][t]^2 - c1\,u[2][t] + 4\,u[0][t]\times u[2][t]$$

Out[83]=
$$-c2\,u[1][t] + 6\,c1\,u[0][t]\times u[1][t] - 30\,u[0][t]^2\,u[1][t] + 2\,u[1][t]\times u[2][t] - c1\,u[3][t] + 4\,u[0][t]\times u[3][t]$$

Out[84]=
$$c1\,c3 + c1\,c2\,u[0][t] - 4\,c3\,u[0][t] - 3\,c1^2\,u[0][t]^2 - 4\,c2\,u[0][t]^2 + 22\,c1\,u[0][t]^3 - 40\,u[0][t]^4 + c1\,u[1][t]^2 - 40\,u[0][t]\,u[1][t]^2 + c1^2\,u[2][t] - c2\,u[2][t] - 8\,c1\,u[0][t]\times u[2][t] + 10\,u[0][t]^2\,u[2][t] + 2\,u[2][t]^2 + 6\,u[1][t]\times u[3][t]$$

In[85]:=
```
nsolt[{u00_, u10_, u20_, u30_}, {c1_, c2_, c3_}, {t0_, t1_}] :=
 NDSolve[
   {u[0]'[t] == -6 u[0][t] × u[1][t] + u[3][t],
    u[1]'[t] ==
     -c3 - c2 u[0][t] + 3 c1 u[0][t]^2 - 10 u[0][t]^3 - u[1][t]^2 - c1 u[2][t] + 4 u[0][t] × u[2][t],
    u[2]'[t] == -c2 u[1][t] + 6 c1 u[0][t] × u[1][t] -
      30 u[0][t]^2 u[1][t] + 2 u[1][t] × u[2][t] - c1 u[3][t] + 4 u[0][t] × u[3][t],
    u[3]'[t] == c1 c3 + c1 c2 u[0][t] - 4 c3 u[0][t] - 3 c1^2 u[0][t]^2 - 4 c2 u[0][t]^2 +
      22 c1 u[0][t]^3 - 40 u[0][t]^4 + c1 u[1][t]^2 - 40 u[0][t] u[1][t]^2 + c1^2 u[2][t] - c2 u[2][t] -
      8 c1 u[0][t] × u[2][t] + 10 u[0][t]^2 u[2][t] + 2 u[2][t]^2 + 6 u[1][t] × u[3][t],
    u[0][t0] == u00,
    u[1][t0] == u10,
    u[2][t0] == u20,
    u[3][t0] == u30},
   {u[0], u[1], u[2], u[3]},
   {t, t0, t1}][[1]]
```

Решаем задачу Коши для (1) от $x = x_0$ до x_1. Значения вектора (u_0, u_1, u_2, u_3) в промежуточных точках, с некоторым шагом δ, принимаем в качестве начальных условий для (2). Решаем соответствующие задачи Коши от $t = t_0$ до t_1.

Потом делаем то же самое в другом порядке, с шагом ϵ по t. Сравниваем получившиеся сетки. Если значения совпадают – значит функция $u(x, t)$ определена корректно. Можно строить её 3D график по полученной сетке.

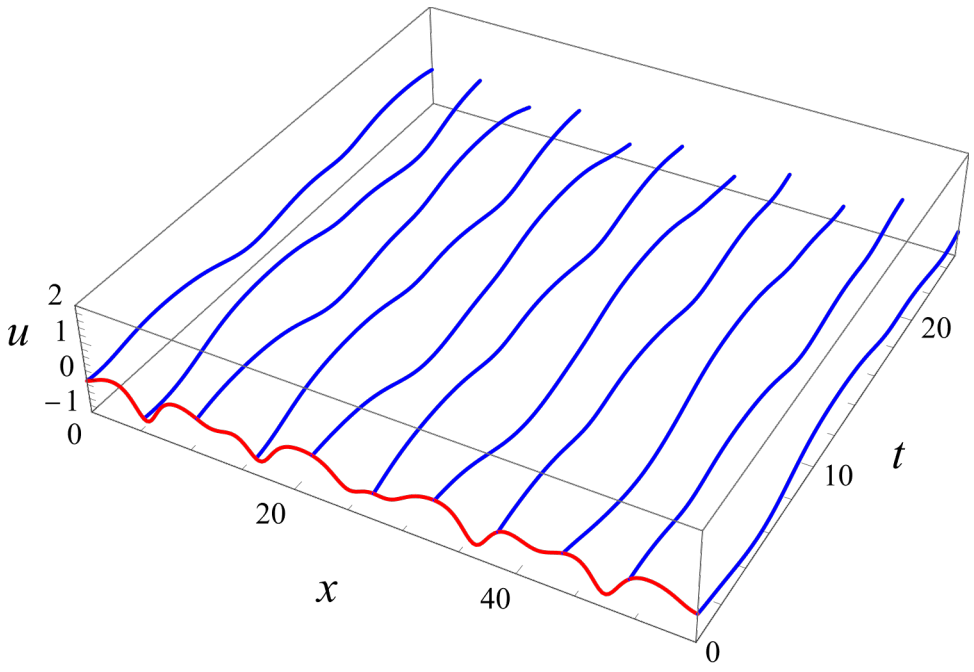
Следует отметить, что в этой задаче многие решения имеют полюсные особенности. Ниже, значения параметров и начальных условий подобраны так, чтобы их не было.

In[86]:=
```
x0 = 0; x1 = 55; (* интервал по x *)
Mx = 200; (* число точек в сетке по x для 3D графика *)
mx = 10; (* число точек для линейчатых графиков *)

t0 = 0; t1 = 25; (* интервал по t *)
Mt = 200; (* число точек в сетке по t для 3D графика *)
mt = 10; (* число точек для линейчатых графиков *)

cc = {0.15, -0.7, 0}; (* параметры *)
U0 = {-0.1, 0.1, 0.1, -0.125}; (* начальные данные при x0, t0 *)
```

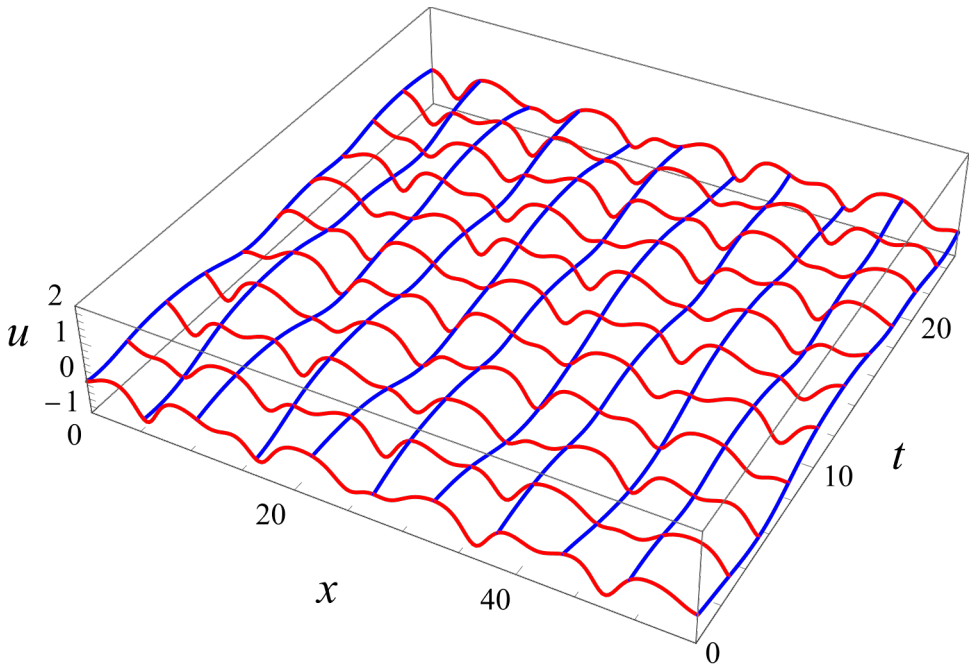
Сначала решим на редкой сетке и построим графики в виде набора кривых. Этого делается просто для наглядности, чтобы получить картинку как на лекции.

In[94]:=
```
(* решаем систему по t;
решаем систему по x для последовательности значений t[0]= t0, ..., t[mt]=t1;
результат сохраняем в виде графического примитива Line *)

nst = nsolt[U0, cc, {t0, t1}];

nstx1 = Table[
    ns = nsolx[
      {u[0][t], u[1][t], u[2][t], u[3][t]} /. t → t0 + j (t1 - t0) / mt /. nst, cc, {x0, x1}];
    Line[Table[
      {x0 + i (x1 - x0) / Mx, t0 + j (t1 - t0) / mt, u[0][x0 + i (x1 - x0) / Mx] /. ns}, {i, 0, Mx}]],
    {j, 0, mt}];

(* делаем то же самое, меняя ролями x и t *)
nsx = nsolx[U0, cc, {x0, x1}];

nsxt1 = Table[
    ns = nsolt[
      {u[0][x], u[1][x], u[2][x], u[3][x]} /. x → x0 + i (x1 - x0) / mx /. nsx, cc, {t0, t1}];
    Line[Table[
      {x0 + i (x1 - x0) / mx, t0 + j (t1 - t0) / Mt, u[0][t0 + j (t1 - t0) / Mt] /. ns}, {j, 0, Mt}]],
    {i, 0, mx}];

gr0 = Graphics3D[{Thick, Red, nstx1[[1]], Blue, nsxt1},
   PlotRange → {{x0, x1}, {t0, t1}, {-1, 2}},
   Axes → True,
   AxesLabel → Evaluate[Style[#, Italic, 24] & /@ {"x", "t", "u"}],
   ImageSize → 500,
   BoxRatios → {1, 1, 0.2},
   BaseStyle → {FontFamily → "Times", FontSize → 16}]

gr1 = Graphics3D[{Thick, Red, nstx1, Blue, nsxt1},
   PlotRange → {{x0, x1}, {t0, t1}, {-1, 2}},
   Axes → True,
   AxesLabel → Evaluate[Style[#, Italic, 24] & /@ {"x", "t", "u"}],
   ImageSize → 500,
   BoxRatios → {1, 1, 0.2},
   BaseStyle → {FontFamily → "Times", FontSize → 16}]
```

Out[98]=



Out[99]=



Красные кривые (эволюция по x) пересекают синие (эволюция по t), что визуально подтверждает коммутативность потоков.

Теперь повторим это же вычисление с более мелким шагом и построим график в виде поверхности.

In[100]:=

```
(* решаем систему по t;
решаем систему по x для последовательности значений t[0]= t0, ..., t[Mt]=t1;
сохраняем результат в виде 2D массива *)

nst = nsolt[U0, cc, {t0, t1}];
nstx2 = Table[
   ns = nsolx[
     {u[0][t], u[1][t], u[2][t], u[3][t]} /. t → t0 + j (t1 - t0) / Mt /. nst, cc, {x0, x1}];
   Table[u[0][x0 + i (x1 - x0) / Mx] /. ns, {i, 0, Mx}], {j, 0, Mt}];

(* делаем то же самое, меняя ролями x и t *)
nsx = nsolx[U0, cc, {x0, x1}];
nsxt2 = Table[
   ns = nsolt[
     {u[0][x], u[1][x], u[2][x], u[3][x]} /. x → x0 + i (x1 - x0) / Mx /. nsx, cc, {t0, t1}];
   Table[u[0][t0 + j (t1 - t0) / Mt] /. ns, {j, 0, Mt}], {i, 0, Mx}];
```

Сравниваем две полученные сетки. Значения, вычисленные двумя способами, совпадают с точностью до численных ошибок, что еще раз подтверждает коммутативность.

In[104]:=

```
Max[Abs[nstx2 - Transpose[nsxt2]]]
```

Out[104]=

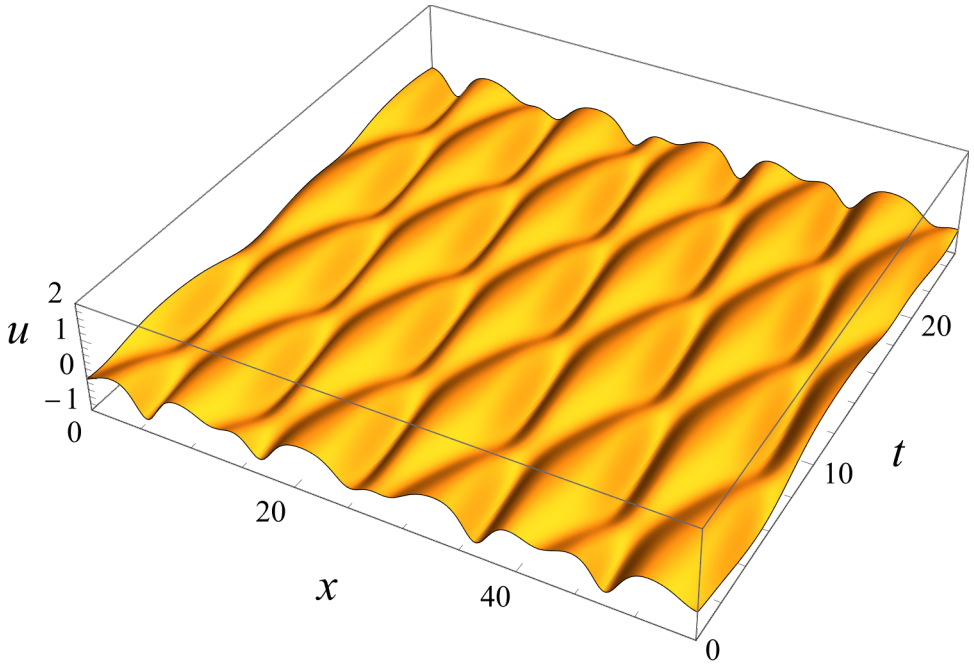
```
0.000922102
```

Теперь можно использовать полученный массив точек (любой из двух), чтобы построить график решения.

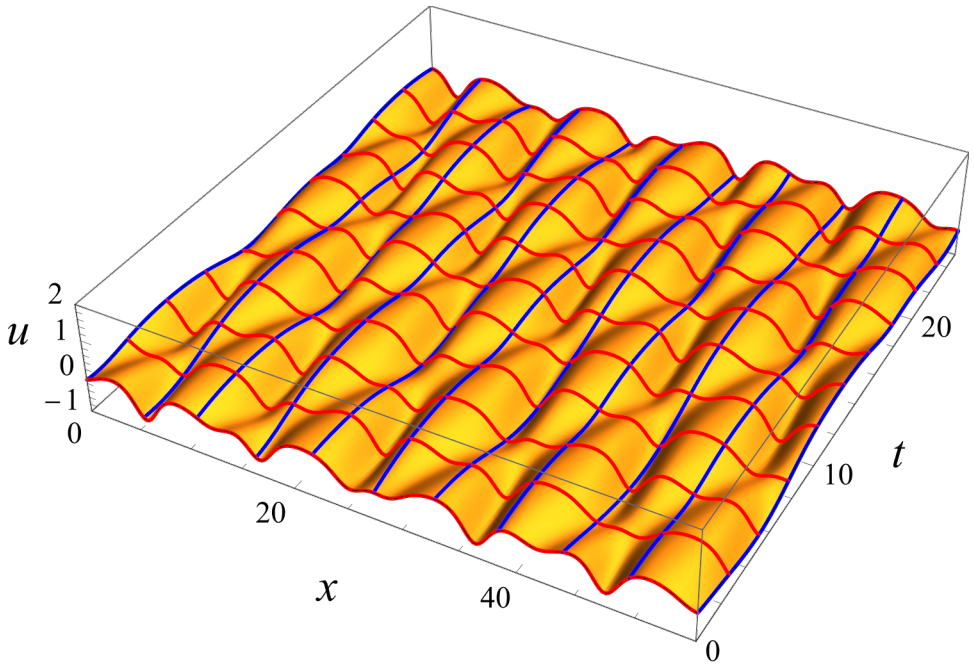
In[105]:=

```
gr3 = ListPlot3D[nstx2,
   DataRange → {{x0, x1}, {t0, t1}},
   PlotRange → {-1, 2},
   Mesh → None,
   AxesLabel → Evaluate[Style[#, Italic, 24] & /@ {"x", "t", "u"}],
   ImageSize → 500,
   BaseStyle → {FontFamily → "Times", FontSize → 16},
   BoxRatios → {1, 1, 0.2},
   PlotRangePadding → 0]

gr2 = Show[gr3, gr1]
```

Out[105]=



Out[106]=



3 Символьные вычисления

Определения из раздела 3.1 используются во всех дальнейших.

3.1 D_x, $f_*(g)$, $E(f)$, интегрирование по частям

Определение D_x, $f_*(g)$, $E(f)$

Некоторые конструкции функционального программирования:
- _u – пример паттерна: всё, что имеет голову u, например u[n], u[x,y], ...
- Cases[f,...] – вхождения в выражение f
- Union – объединение элементов списка
- лямбда-функция: (выражение с #)&
- f/@xxx – применение f к каждому элементу списка xxx
- y@@xxx – замена головы выражения xxx на y (в нашем случае List заменяется на Plus)
- Nest[f,x,n] – применяет f к x n раз

In[107]:=
```
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]
```

Теперь можем определить эволюционную производную

$$f_*(g) = \nabla_g(f) = \partial_0(f) + \ldots + \partial_n(f)\, D_x^n(g),$$

и вариационную производную (оператор Эйлера)

$$E(f) = \frac{\delta f}{\delta u} = \partial_0(f) - D_x(\partial_1(f)) + D_x^2(\partial_2(f)) - \ldots + (-D_x)^n(\partial_n(f)),$$

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

In[111]:=
```
star[g_, f_] := dif[g] /. dd[u[k_]] :> dx[f, k]
vard[f_] := Expand[Plus @@ ((-1) ^ #[[1]] dx[D[f, #], #[[1]]] & /@ vars[f])]
```

Утверждение: выражение является полной производной по x тогда и только тогда, когда его вариационная производная равна 0:

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

Алгоритм интегрирования по частям

Некоторые конструкции процедурного программирования:
- If[cond, t, f, u] t, f, u – результаты, если cond = True, False, undefined
- And[cond_1, cond_2, ...] = cond_1 && cond_2 && ...
- Module[{локальные переменные}, команды]
- While[cond, команды]

In[113]:=

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

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} (* возвращаем пару (a,b)- результат и препятствие *)
  ]
 ]
```

Протестируем. Определим какое-нибудь не слишком сложное выражение (многочлен).

In[115]:=

```
f = Expand[(u[0] + u[1]^2 + 3 x)^2 + 2 u[3] × u[4] Exp[u[0]]]
vard[f]
int[f]
```

Out[115]=

$$9 x^2 + 6 x u[0] + u[0]^2 + 6 x u[1]^2 + 2 u[0] u[1]^2 + u[1]^4 + 2 e^{u[0]} u[3] \times u[4]$$

Out[116]=

$$6 x + 2 u[0] - 12 u[1] - 2 u[1]^2 - 12 x u[2] - 4 u[0] \times u[2] - 12 u[1]^2 u[2] + 2 e^{u[0]} u[1]^4 u[3] + 12 e^{u[0]} u[1]^2 u[2] \times u[3] + 6 e^{u[0]} u[2]^2 u[3] + 8 e^{u[0]} u[1] u[3]^2 + 6 e^{u[0]} u[1]^3 u[4] + 18 e^{u[0]} u[1] \times u[2] \times u[4] + 10 e^{u[0]} u[3] \times u[4] + 6 e^{u[0]} u[1]^2 u[5] + 6 e^{u[0]} u[2] \times u[5] + 2 e^{u[0]} u[1] \times u[6]$$

Out[117]=

$$\left\{e^{u[0]} u[3]^2,\ 9 x^2 + 6 x u[0] + u[0]^2 + 6 x u[1]^2 + 2 u[0] u[1]^2 + u[1]^4 - e^{u[0]} u[1] u[3]^2\right\}$$

Как видим, $f \notin \operatorname{Im} D_x$. Сделаем проверку для dx[f].

In[118]:=

```
Expand[dx[f]]
vard[%]
int[%%]
%[[1]] - f
```

Out[118]=

$$18 x + 6 u[0] + 6 x u[1] + 2 u[0] \times u[1] + 6 u[1]^2 + 2 u[1]^3 + 12 x u[1] \times u[2] + 4 u[0] \times u[1] \times u[2] + 4 u[1]^3 u[2] + 2 e^{u[0]} u[1] \times u[3] \times u[4] + 2 e^{u[0]} u[4]^2 + 2 e^{u[0]} u[3] \times u[5]$$

Out[119]=

$$0$$

Out[120]=

$$\left\{9 x^2 + 6 x u[0] + u[0]^2 + 6 x u[1]^2 + 2 u[0] u[1]^2 + u[1]^4 + 2 e^{u[0]} u[3] \times u[4],\ 0\right\}$$

Out[121]=

$$0$$

3.2 Законы сохранения КдФ (обращение преобразования Миуры)

В лекции 2 было доказано, что ряд

$$v = -z + V_0 + V_1/(2z) + V_2/(2z)^2 + ...,$$

удовлетворяющий уравнению

$$v_x + v^2 = z^2 + u,$$

служит производящей функцией для плотностей законов сохранения уравнения КдФ

(1) $u_t = u_3 - 6\,u\,u_1$.

Для коэффициентов получаются рекуррентные соотношения

$$V_0 = 0,\ V_1 = -u,$$

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

Вычислим несколько первых коэффициентов.

In[122]:=

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

Чтобы было удобнее смотреть, заменим $u[n]$ на u_n:

In[125]:=

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

$$V_1 = -u_0$$

$$V_2 = -u_1$$

$$V_3 = u_0^2 - u_2$$

$$V_4 = 4\,u_0\,u_1 - u_3$$

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

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

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

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

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

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

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

$$V_{12} = 1024\,u_0^5\,u_1 - 9600\,u_0^2\,u_1^3 - 11\,520\,u_0^3\,u_1\,u_2 + 13\,560\,u_1^3\,u_2 + 24\,480\,u_0\,u_1\,u_2^2 - 1280\,u_0^4\,u_3 + 18\,000\,u_0\,u_1^2\,u_3 + 10\,880\,u_0^2\,u_2\,u_3 - 9524\,u_2^2\,u_3 - 7000\,u_1\,u_3^2 + 6400\,u_0^2\,u_1\,u_4 - 11\,140\,u_1\,u_2\,u_4 - 5000\,u_0\,u_3\,u_4 + 640\,u_0^3\,u_5 - 2720\,u_1^2\,u_5 - 3320\,u_0\,u_2\,u_5 + 922\,u_4\,u_5 - 1400\,u_0\,u_1\,u_6 + 658\,u_3\,u_6 - 160\,u_0^2\,u_7 + 328\,u_2\,u_7 + 108\,u_1\,u_8 + 20\,u_0\,u_9 - u_{11}$$

$$V_{13} = -132\,u_0^7 + 11\,550\,u_0^4\,u_1^2 - 24\,310\,u_0\,u_1^4 + 2772\,u_0^5\,u_2 - 87\,516\,u_0^2\,u_1^2\,u_2 - 17\,556\,u_0^3\,u_2^2 + 69\,006\,u_1^2\,u_2^2 + 27\,764\,u_0\,u_2^3 - 25\,872\,u_0^3\,u_1\,u_3 + 33\,760\,u_1^3\,u_3 + 122\,408\,u_0\,u_1\,u_2\,u_3 + 13\,662\,u_0^2\,u_3^2 - 26\,322\,u_2\,u_3^2 - 2310\,u_0^4\,u_4 + 35\,860\,u_0\,u_1^2\,u_4 + 21\,780\,u_0^2\,u_2\,u_4 - 20\,922\,u_2^2\,u_4 - 30\,776\,u_1\,u_3\,u_4 - 5522\,u_0\,u_4^2 + 10\,692\,u_0^2\,u_1\,u_5 - 20\,376\,u_1\,u_2\,u_5 - 9196\,u_0\,u_3\,u_5 + 923\,u_5^2 + 924\,u_0^3\,u_6 - 4270\,u_1^2\,u_6 - 5236\,u_0\,u_2\,u_6 + 1582\,u_4\,u_6 - 1936\,u_0\,u_1\,u_7 + 988\,u_3\,u_7 - 198\,u_0^2\,u_8 + 438\,u_2\,u_8 + 130\,u_1\,u_9 + 22\,u_0\,u_{10} - u_{12}$$

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

■ Series[f, {z, z_0, n}] разложение в ряд Тейлора или Лорана

In[126]:=

```
v = -z + Sum[V[n] / (2 z)^n, {n, 1, M}];
Series[Expand[v^2 + dx[v] - z^2 - u[0]], {z, Infinity, M - 1}]
```

Out[127]=

$$O\left[\frac{1}{z}\right]^{13}$$

Теория утверждает, что:

(i) $V_{2\,n} = D_x(a_n)$ (тривиальные плотности)

(ii) $D_t(V_{2\,n-1}) = D_x(b_n)$ и $V_{2\,n-1} \notin \mathrm{Im}\, D_x$

для некоторых a_n, b_n – функций от u, u_1, u_2, … Проверяем:

In[128]:=
```
Table[vard[V[n]], {n, 2, M, 2}]
Table[vard[V[n]] === 0, {n, 1, M, 2}]
Table[vard[star[V[n], u[3] - 6 u[0]×u[1]]], {n, 1, M, 2}]
```

Out[128]=
```
{0, 0, 0, 0, 0, 0}
```

Out[129]=
```
{False, False, False, False, False, False, False}
```

Out[130]=
```
{0, 0, 0, 0, 0, 0, 0}
```

Функции b_n (токи законов сохранения) можно вычислить, применяя интегрирование по частям. Например:

In[131]:=
```
ρ = V[11]
Expand[star[%, u[3] - 6 u[0]×u[1]]]
int[%]
σ = %[[1]];

Expand[star[ρ, u[3] - 6 u[0]×u[1]] - dx[σ]]
```

Out[131]=
```
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]
```

Out[132]=
```
-1512 u[0]^6 u[1] + 63 000 u[0]^3 u[1]^3 - 26 520 u[1]^5 + 51 660 u[0]^4 u[1]×u[2] - 312 936 u[0] u[1]^3 u[2] -
 273 888 u[0]^2 u[1] u[2]^2 + 168 300 u[1] u[2]^3 + 4032 u[0]^5 u[3] - 202 212 u[0]^2 u[1]^2 u[3] -
 77 616 u[0]^3 u[2]×u[3] + 372 828 u[1]^2 u[2]×u[3] + 224 400 u[0] u[2]^2 u[3] +
 165 828 u[0]×u[1] u[3]^2 - 26 322 u[3]^3 - 45 528 u[0]^3 u[1]×u[4] + 72 880 u[1]^3 u[4] +
 263 256 u[0]×u[1]×u[2]×u[4] + 58 032 u[0]^2 u[3]×u[4] - 125 264 u[2]×u[3]×u[4] -
 36 800 u[1] u[4]^2 - 3150 u[0]^4 u[5] + 64 392 u[0] u[1]^2 u[5] + 38 376 u[0]^2 u[2]×u[5] -
 41 298 u[2]^2 u[5] - 61 184 u[1]×u[3]×u[5] - 22 080 u[0]×u[4]×u[5] + 16 164 u[0]^2 u[1]×u[6] -
 34 628 u[1]×u[2]×u[6] - 15 744 u[0]×u[3]×u[6] + 3428 u[5]×u[6] + 1176 u[0]^3 u[7] -
 6382 u[1]^2 u[7] - 7824 u[0]×u[2]×u[7] + 2570 u[4]×u[7] - 2580 u[0]×u[1]×u[8] +
 1426 u[3]×u[8] - 234 u[0]^2 u[9] + 568 u[2]×u[9] + 154 u[1]×u[10] + 24 u[0]×u[11] - u[13]
```

Out[133]=
```
{-216 u[0]^7 + 15 750 u[0]^4 u[1]^2 - 26 520 u[0] u[1]^4 + 4032 u[0]^5 u[2] - 103 428 u[0]^2 u[1]^2 u[2] -
  22 344 u[0]^3 u[2]^2 + 69 006 u[1]^2 u[2]^2 + 30 288 u[0] u[2]^3 - 32 928 u[0]^3 u[1]×u[3] +
  33 760 u[1]^3 u[3] + 133 536 u[0]×u[1]×u[2]×u[3] + 16 146 u[0]^2 u[3]^2 -
  26 322 u[2] u[3]^2 - 3150 u[0]^4 u[4] + 39 120 u[0] u[1]^2 u[4] + 25 740 u[0]^2 u[2]×u[4] -
  20 922 u[2]^2 u[4] - 30 776 u[1]×u[3]×u[4] - 6024 u[0] u[4]^2 + 12 636 u[0]^2 u[1]×u[5] -
  20 376 u[1]×u[2]×u[5] - 10 032 u[0]×u[3]×u[5] + 923 u[5]^2 + 1176 u[0]^3 u[6] -
  4270 u[1]^2 u[6] - 5712 u[0]×u[2]×u[6] + 1582 u[4]×u[6] - 2112 u[0]×u[1]×u[7] +
  988 u[3]×u[7] - 234 u[0]^2 u[8] + 438 u[2]×u[8] + 130 u[1]×u[9] + 24 u[0]×u[10] - u[12], 0}
```

Out[135]=
```
0
```

3.3 Построение и проверка высших симметрий КдФ. Оператор рекурсии

Напомним (см. лекцию 4), что уравнение КдФ (1) допускает бесконечную последовательность высших симметрий нечётного порядка по производным

(2) $u_{t_k} = f_k(u, \ ..., u_{2\,k-1}), \quad k = 1, 2 \ldots ,$

то есть, таких уравнений, что $[D_t, D_{t_k}] = 0$. В частности,

$$u_{t_1} = f_1 = u_1,$$
$$u_{t_2} = f_2 = u_3 - 6\,u\,u_1,$$
$$u_{t_3} = f_3 = u_5 - 10\,u\,u_3 - 20\,u_1\,u_2 + 30\,u^2\,u_1,$$
$$u_{t_4} = f_4 = u_7 - 14\,u\,u_5 - 42\,u_1\,u_4 - 70\,u_2\,u_3 + 70\,u^2\,u_3 + 280\,u\,u_1\,u_2 + 70\,u_1^3 - 140\,u^3\,u_1,$$

и так далее. Первые две симметрии классические, то есть, отвечают однопараметрическим группам точечных преобразований (сдвиги $x \to x + a$ и $t \to t + a$). Следующие симметрии – высшие.

Есть несколько способов вычисления f_k:

(i) исходя из определения симметрии;
(ii) метод неопределённых коэффициентов, с использованием однородности $w(u) = 2, \ w(\partial_x) = 1$;
(iii) при помощи мастер-симметрии;
(iv) оператор рекурсии (лекция 4);
(v) квадратичное рекуррентное соотношение (лекция 4);
(vi) расширение рекуррентного соотношения для законов сохранения.

Способы (iv)–(vi) примерно эквивалентны и основаны на рекуррентном соотношении

(3) $D_x(f_{k+1}) = \left(D_x^3 - 4\,u\,D_x - 6\,u_1\right)(f_k).$

Применяя D_x^{-1}, можно привести это к виду

$$f_{k+1} = R(f_k), \quad R = D_x^2 - 4\,u - 2\,u_1\,D_x^{-1}.$$

Вычислим симметрии до f_{15} за 1.5 сек.

- Timing[expr] возвращает {затраченное время, результат}
- в нашем случае результат пуст, так как цикл Do ничего не возвращает, только что-то делает внутри себя
- перед обращением к Timing очищаем f и кэш, так как иначе при повторном запуске ячейки время окажется заниженным – *Mathematica* не будет считать заново, а воспользуется сохраненными в памяти результатами.

In[136]:=

```
R[f_] := Expand[dx[f, 2] - 4 u[0] f - 2 u[1] × int[f]〚1〛]

M = 15;
Clear[f]
ClearSystemCache[]

Timing[
 f[1] = u[1];
 Do[f[j] = R[f[j - 1]], {j, 2, M}]]
```

Out[140]=

```
{1.32813, Null}
```

Посмотрим, что получилось.

In[141]:=

```
Do[Print[f[j] /. u[n_] :> u_n], {j, 1, 7}]

Length /@ Table[f[j], {j, 1, M}] (* OEIS A182747 *)
```

u_1

$-6\,u_0\,u_1 + u_3$

$30\,u_0^2\,u_1 - 20\,u_1\,u_2 - 10\,u_0\,u_3 + u_5$

$-140\,u_0^3\,u_1 + 70\,u_1^3 + 280\,u_0\,u_1\,u_2 + 70\,u_0^2\,u_3 - 70\,u_2\,u_3 - 42\,u_1\,u_4 - 14\,u_0\,u_5 + u_7$

$630\,u_0^4\,u_1 - 1260\,u_0\,u_1^3 - 2520\,u_0^2\,u_1\,u_2 + 1302\,u_1\,u_2^2 - 420\,u_0^3\,u_3 + 966\,u_1^2\,u_3 + 1260\,u_0\,u_2\,u_3 + 756\,u_0\,u_1\,u_4 - 252\,u_3\,u_4 + 126\,u_0^2\,u_5 - 168\,u_2\,u_5 - 72\,u_1\,u_6 - 18\,u_0\,u_7 + u_9$

$-2772\,u_0^5\,u_1 + 13\,860\,u_0^2\,u_1^3 + 18\,480\,u_0^3\,u_1\,u_2 - 14\,784\,u_1^3\,u_2 - 28\,644\,u_0\,u_1\,u_2^2 + 2310\,u_0^4\,u_3 - 21\,252\,u_0\,u_1^2\,u_3 - 13\,860\,u_0^2\,u_2\,u_3 + 9702\,u_2^2\,u_3 + 7194\,u_1\,u_3^2 - 8316\,u_0^2\,u_1\,u_4 + 11\,484\,u_1\,u_2\,u_4 + 5544\,u_0\,u_3\,u_4 - 924\,u_0^3\,u_5 + 2838\,u_1^2\,u_5 + 3696\,u_0\,u_2\,u_5 - 924\,u_4\,u_5 + 1584\,u_0\,u_1\,u_6 - 660\,u_3\,u_6 + 198\,u_0^2\,u_7 - 330\,u_2\,u_7 - 110\,u_1\,u_8 - 22\,u_0\,u_9 + u_{11}$

$12\,012\,u_0^6\,u_1 - 120\,120\,u_0^3\,u_1^3 + 30\,030\,u_1^5 - 120\,120\,u_0^4\,u_1\,u_2 + 384\,384\,u_0\,u_1^3\,u_2 + 372\,372\,u_0^2\,u_1\,u_2^2 - 177\,320\,u_1\,u_2^3 - 12\,012\,u_0^5\,u_3 + 276\,276\,u_0^2\,u_1^2\,u_3 + 120\,120\,u_0^3\,u_2\,u_3 - 392\,964\,u_1^2\,u_2\,u_3 - 252\,252\,u_0\,u_2^2\,u_3 - 187\,044\,u_0\,u_1\,u_3^2 + 26\,598\,u_3^3 + 72\,072\,u_0^3\,u_1\,u_4 - 77\,220\,u_1^3\,u_4 - 298\,584\,u_0\,u_1\,u_2\,u_4 - 72\,072\,u_0^2\,u_3\,u_4 + 126\,984\,u_2\,u_3\,u_4 + 37\,466\,u_1\,u_4^2 + 6006\,u_0^4\,u_5 - 73\,788\,u_0\,u_1^2\,u_5 - 48\,048\,u_0^2\,u_2\,u_5 + 42\,042\,u_2^2\,u_5 + 62\,348\,u_1\,u_3\,u_5 + 24\,024\,u_0\,u_4\,u_5 - 20\,592\,u_0^2\,u_1\,u_6 + 35\,464\,u_1\,u_2\,u_6 + 17\,160\,u_0\,u_3\,u_6 - 3432\,u_5\,u_6 - 1716\,u_0^3\,u_7 + 6578\,u_1^2\,u_7 + 8580\,u_0\,u_2\,u_7 - 2574\,u_4\,u_7 + 2860\,u_0\,u_1\,u_8 - 1430\,u_3\,u_8 + 286\,u_0^2\,u_9 - 572\,u_2\,u_9 - 156\,u_1\,u_{10} - 26\,u_0\,u_{11} + u_{13}$

Out[142]=

```
{1, 2, 4, 8, 14, 24, 41, 66, 105, 165, 253, 383, 574, 847, 1238}
```

Проверим, что это действительно симметрии.

In[143]:=

```
ClearSystemCache[]
Timing[
 Table[star[f[2], f[j]] - star[f[j], f[2]], {j, 1, M}] // Expand]
```

Out[144]=

```
{6.625, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}}
```

Проверим, что и друг с другом они коммутируют

In[145]:=

```
Table[star[f[i], f[j]] - star[f[j], f[i]], {i, 1, 5}, {j, i + 1, 6}] // Expand
```

Out[145]=

```
{{0, 0, 0, 0, 0}, {0, 0, 0, 0}, {0, 0, 0}, {0, 0}, {0}}
```

3.4 Явное рекуррентное соотношение

Пусть $f_k = D_x(a_k)$ и $a = 1/2 + a_1/(-4\lambda) + a_2/(-4\lambda)^2 + \ldots$ – производящая функция для a_k. Тогда рекуррентное соотношение (3) сводится к

$$2\,a\,a_{xx} - a_x^2 + 4\,(\lambda - u)\,a^2 = \lambda.$$

В свою очередь, это сводится (проверьте!) к

$$v_x + v^2 = z^2 + u, \quad a(\lambda)\,(v(-z) - v(z)) = z, \quad z^2 = -\lambda.$$

Как мы уже знаем, первое уравнение даёт рекуррентное соотношение для плотностей законов сохранения, второе просто сводится к вычислению обратного ряда. Вычислим те же симметрии, что и раньше, и
проверим совпадение результатов.

In[146]:=

```
Clear[a, V, ff]
ClearSystemCache[];

Timing[
 V[1] = -u[0];
 Do[V[n + 1] = Factor[(dx[V[n]] + Sum[V[s] × V[n - s], {s, 1, n - 1}])], {n, 1, 2 M - 2}];
 a[0] = -1 / 2;
 Do[a[k] = Expand[-V[2 k - 1] + 2 Sum[a[k - s] × V[2 s - 1], {s, 1, k - 1}]];
  ff[k] = dx[a[k]], {k, 1, M}]
]

Table[Expand[ff[k] - f[k]], {k, 1, M}]
```

Out[148]=

```
{1.03125, Null}
```

Out[149]=

```
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
```