

Построение солитонов КдФ при помощи суперпозиции преобразований Бэклунда

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

Построение n -солитонного решения уравнения КдФ

$$u = 2 \partial_x^2 \log W(y_1, \dots, y_n), \quad y_j = e^{X_j} + (-1)^{n-j} e^{-X_j}, \quad X_j = k_j x + 4 k_j^3 t + \delta_j, \quad 0 < k_n < \dots < k_1,$$

по схеме, описанной в [10_Backlund transformations.pdf](#).

In[49]:=

```
(* число солитонов *)
n = 10;

(* задаем случайные собственные значения и сортируем их по убыванию,
также задаем случайные сдвиги *)
K = Sort[RandomReal[{1/2, 3/2}, n], #1 > #2 &]
d = RandomReal[{-5, 5}, n]

(* фазы *)
X[j_] := K[[j]] x + 4 K[[j]]^3 t + d[[j]]

(* затравочные решения *)
q[] = 0;
Do[q[j] = -K[[j]] Tanh[X[j]], {j, n, 1, -2}]
Do[q[j] = -K[[j]] Coth[X[j]], {j, n - 1, 1, -2}]

(* формула суперпозиции *)
q[i___, j_, k_] := q[i] + (K[[j]]^2 - K[[k]]^2) / (q[i, j] - q[i, k])

(* вычисление решения *)
u = -2 D[q @@ Range[n], x];

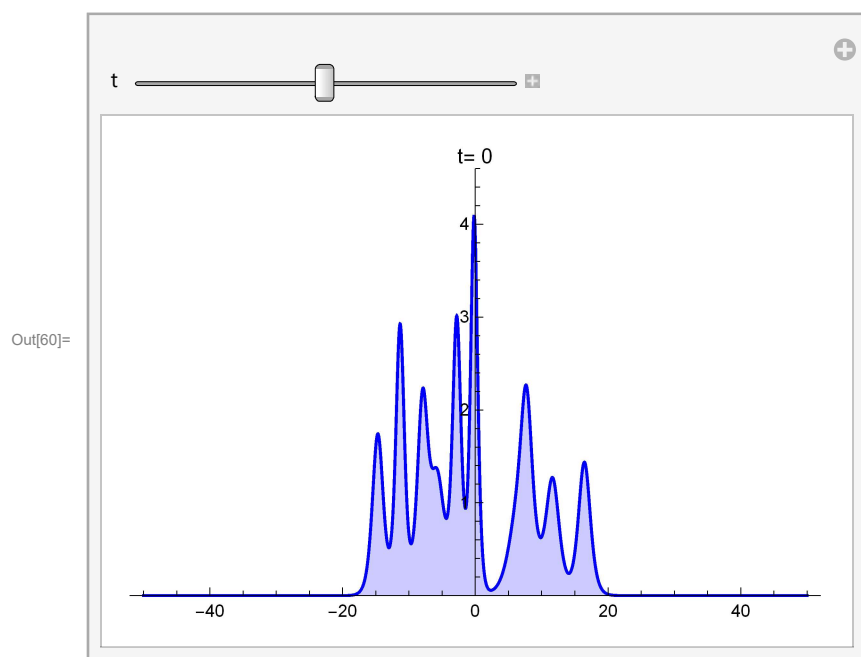
(* компилированная функция, это ускоряет счет *)
cu = Compile[{{x, _Real}, {t, _Real}}, Evaluate[u]];

(* построение графика *)
cgrou[t_, xmin_, xmax_] := Plot[cu[x, t], {x, xmin, xmax},
  PlotRange -> {0, 2.1 K[[1]]^2},
  PlotStyle -> Blue,
  Filling -> 0,
  PlotPoints -> 50,
  PlotLabel -> "t= " <> ToString[t]];

(* анимация по t *)
Manipulate[cgrou[t, -50, 50],
  {{t, 0}, -10, 10, Appearance -> "Labelled"}]
```

Out[50]= {1.4799, 1.26654, 1.25404, 1.20446, 1.17185, 0.931983, 0.879084, 0.825972, 0.72158, 0.547984}

Out[51]= {2.39118, 4.71256, 4.40188, -1.87971, 3.97077, 4.04218, -4.39865, -4.03362, 1.10692, -1.85851}



По такой схеме можно построить без особого труда 10–12 солитонов, (с вронскианами – не больше 4). При большем числе и здесь счет становится медленным, а также накапливаются ошибки в арифметических операциях. Если немного постараться, то можно дойти до 15, но дальше *Mathematica* справляется плохо. На курсе 2018, Д. Чукланов довел число солитонов до 40, с использованием Си.