

УДК 519.711.3+551.5.001.57+517.957

Ю. Г. Игнатьев¹, А. Р. Самигуллина²

ЧИСЛЕННО-АНАЛИТИЧЕСКИЕ МЕТОДЫ МАТЕМАТИЧЕСКОГО МОДЕЛИРОВАНИЯ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ В СКМ MAPLE.

III. ВИЗУАЛИЗАЦИЯ НЕЛИНЕЙНЫХ ДИНАМИЧЕСКИХ СИСТЕМ И ПРИЛОЖЕНИЯ К МЕХАНИКЕ И КОСМОЛОГИИ

Описаны инструменты автоматической визуализации численных решений систем нелинейных обыкновенных дифференциальных уравнений и усреднения численных решений на основе авторского прикладного пакета программ DifEqTools, а также приведены примеры применения этих инструментов к исследованию нелинейных задач механики и космологии.

Ключевые слова: нелинейные динамические системы, математическое моделирование, качественная теория дифференциальных уравнений, сплайновая аппроксимация, численные методы интегрирования, визуализация вычислений, нелинейная механика, космология, прикладные математические пакеты.

PACS: 02.60.Cb, 07.05.Tr, 04.25.-g

This work was founded by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities.

1. Введение

В предыдущих работах [1], [2] мы описали некоторые *численно - аналитические методы* исследования нелинейных динамических систем высокой размерности, а в [3], [4] описали соответствующие программные средства исследования таких систем в специальном авторском пакете DifEqTools, являющейся приложением к прикладному математическому пакету Maple. В частности, в [2] описаны автоматизированные средства математического анализа численных решений, получаемых в указанном пакете. В этой статье мы опишем автоматизированные средства построения графиков численных решений нелинейных систем ОДУ, программные процедуры усреднения быстро осциллирующих численных решений, процедуры автоматизированной динамической визуализации математических моделей нелинейных динамических систем на основе сплайновой интерполяции численных решений. Кроме того, мы приведем примеры исследования космологических моделей на основе развитых средств математического и компьютерного моделирования.

2. Программные процедуры автоматизированной визуализации нелинейных динамических систем в пакете DifEqTools

2.1. Сервисные программные процедуры автоматизированного построения графиков функций численных решений

Пусть задана задача Коши для системы нелинейных обыкновенных дифференциальных уравнений, разрешенных относительно старших производных:

$$\begin{aligned} y_i^{(n_i)} &= F_i(y_1, \dots, y_N, y_1', \dots, y_N', y_1'', \dots, y_N'', \dots, y_1^{(n_1-1)}, \dots, y_N^{(n_N-1)}, t); \quad (i = \overline{1, N}), \\ \Rightarrow y_i^{(n_i)} &= F_i(t), \end{aligned} \quad (1)$$

где $y^{(n)} = d^n y / dt^n$ — обозначение n -той производной функции $y(t)$ по независимой переменной t , временной переменной³, а F_i — непрерывно-дифференцируемые функции своих переменных. Будем в дальней-

¹E-mail: ignatev_yu@rambler.ru

²E-mail: alsu_sam@rambler.ru

³Хотя это может быть и не обязательно физическое время.

шем полагать выполненными начальные условия для системы (1):

$$y_i^{(k)}(t) \Big|_{t=t_0} = C_i^k; \quad (k = \overline{1, n_i - 1}; i = \overline{1, N}), \quad (2)$$

соответствующие задаче Коши, где $C_i^k \equiv y_i^{(k)}(t_0)$ – начальные значения производных k -го порядка функций $y_i(t)$.

Договоримся в дальнейшем задавать систему ОДУ и соответствующие начальные условия в виде упорядоченных списков:

$$\text{Eqs} = [y_1^{(n_1)} = F_1(t), y_2^{(n_2)} = F_2(t), \dots, y_N^{(n_N)} = F_N(t)]; \quad (3)$$

$$\text{ICS} = [y_1^{(0)}(t_0) = C_1^0, y_1^{(1)}(t_0) = C_1^1, \dots, y_N^{(n_N)}(t_0) = C_N^{n_N}]. \quad (4)$$

Пусть далее `[Method]` есть упорядоченный список опций метода численного интегрирования системы ОДУ в пакете Maple, `i` – номер неизвестной функции y_i в списке (3), `j` – порядок выводимой производной этой функции, `L` – интервал значений независимой переменной t , на котором строится график выводимой функции, `C` – цвет линии. Тогда график функции $y_i^{(k)}(t)$ получается численным интегрированием с помощью программной процедуры

`DifEqTools[GraphConf1](Eqs, ICS, Method, [i, j], L, Color)`.

Приведем пример численного решения с автоматизированным выводом графика системы двух линейных ОДУ второго порядка.

```
> Eqs := [(D@@2)(x)(t) = -0.1*D(y)(t) - x(t), (D@@2)(y)(t) = -0.1*D(x)(t) - y(t)];
```

$$\text{Eqs} := [D^{(2)}(x)(t) = -0.1D(y)(t) - x(t), D^{(2)}(y)(t) = -0.1D(x)(t) - y(t)] \quad (5)$$

```
> ICS := [x(0)=2, D(x)(0)=1, y(0)=0, D(y)(0)=0];
```

$$\text{ICS} := [x(0) = 2, D(x)(0) = 1, y(0) = 0, D(y)(0) = 0] \quad (6)$$

Заметим, во-первых, что хотя мы демонстрируем работу пакета на примере численного решения линейной системы ОДУ, фактор линейности системы при отсутствии сингулярных точек не играет никакой роли при нахождении численного решения. В дальнейшем мы в этом убедимся.

Во-вторых, ряд программных процедур в последующих примерах сопровождается окружением `T:=time():Процедура;time()-T`, позволяющим выводить полное время Δt исполнения операций. Это время, конечно, в значительной степени зависит от особенностей компьютера, версии Maple и операционной системы. Тем не менее, оно позволяет ориентироваться пользователю в выборе опций программных процедур для оптимизации процесса компьютерного моделирования.

В дальнейшем под стандартным компьютером (СК) мы будем подразумевать следующую конфигурацию: CR: Intel(R) Core(TM) i7-3770 CPU@3,49GHz 3,40 GHz, ОЗУ 16,0 ГБ; 64-разрядная ОС Windows 7 Максимальная; Maple 18.

Наконец, в - третьих, многие программные процедуры пакета `DifEqTools` содержат кроме *обязательных параметров*, задаваемых в виде строго упорядоченной последовательности `[par1, par2, ...]`, также и *необязательные параметры*, задаваемых в виде произвольно упорядоченной последовательности `{par1=op1, par2=op2, ...}`, при этом имена необязательных параметров `pari` должны совпадать с именами, назначенными им в пакете Maple, а значения опций `opi` лежать в диапазоне значений, установленных используемой версией Maple (см. справки `Help/Maple Help`). Если необязательные параметры не указаны, пакет реализует их значения, назначенные по умолчанию.

```
> T:=time():
```

```
DifEqTools[GraphConf1](Eqs, ICS, [method=rkf45], [2, 0], 32*Pi, navy);
time()-T;
```

```

> ICS_1:=[x(0)=1,D(x)(0)=1,y(0)=0,D(y)(0)=0];

> T:=time():
plots[display](DifEqTools[GraphConf1](Eqs,ICS_1,[method=rkf45],
[2,0],32*Pi,red),
DifEqTools[GraphConf1](Eqs,ICS,[method=rkf45],[2,0],32*Pi,navy),axes=BOXED);
time()-T;

```

В первом случае время вывода графика составило на СК 0,17 сек, во втором — 0,32 сек.

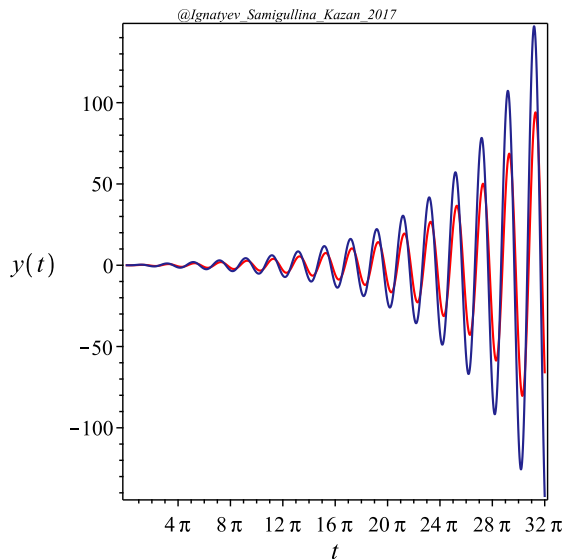


Рис. 1. Автоматизированный вывод численного решения задачи Коши (5) — (6): выводится нулевая производная второй неизвестной функции $y(t)$ на отрезке $[0, 32\pi]$.

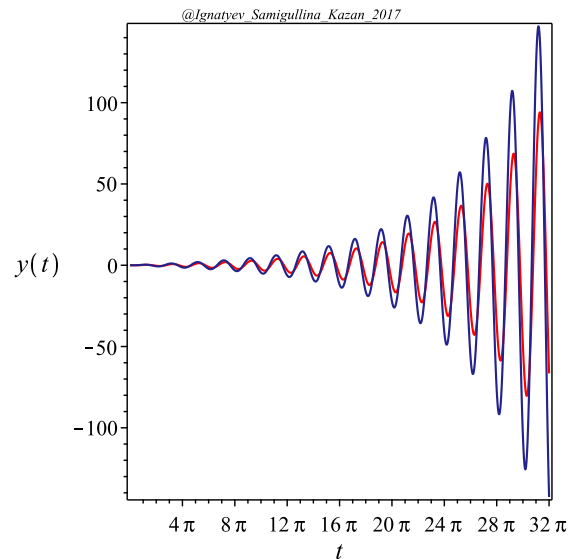


Рис. 2. Совмещение графиков двух численных решений задачи Коши с начальными условиями ICS и ICS₁ с помощью команды `plots[display]`.

Продemonстрируем влияние параметра L — интервала значений независимой переменной при выводе графика численного решения. Рассмотрим следующую задачу Коши для одиночного ОДУ 3-го порядка:

```

> EQS1:=[(D@@3)(x)(t)=-0.1*(D@@2)(x)(t)-0.1*D(x)(t)-0.1*x(t)];
> ICS1:=[x(0)=0,D(x)(0)=0.001,(D@@2)(x)(0)=0];

```

$$EQS1 := \left[D^{(3)}(x)(t) = -0.1 * D^{(2)}(x)(t) - 0.1 * D(x)(t) - 0.1 * x(t) \right] \quad (7)$$

$$EQS1 := \left[x(0) = 0, D(x)(0) = 0.001, D^{(2)}(x)(0) = 0 \right] \quad (8)$$

Сначала выведем решение на всем интервале, начиная от начальной точки — при этом параметру L просто можно придать значение t в конечной точке, а затем придадим значение L на интервале $[600, 630]$:

```

> DifEqTools[GraphConf1](EQS1,ICS1,[method=rkf45],[1,0],1600,navy);
> DifEqTools[GraphConf1](EQS1,ICS1,[method=rkf45],[1,0],[600,630],navy);

```

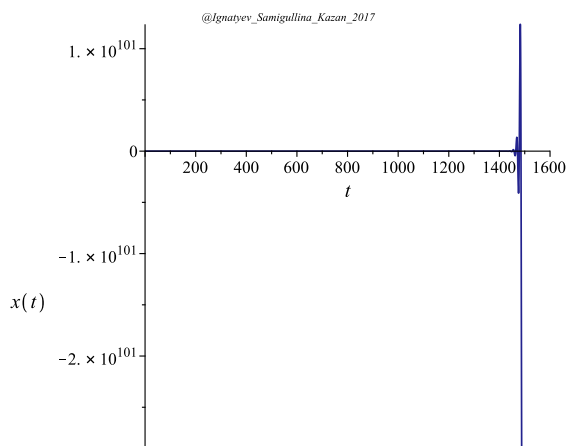


Рис. 3. Автоматизированный вывод численного решения задачи Коши (7) – (8) при параметре $L=1600$.

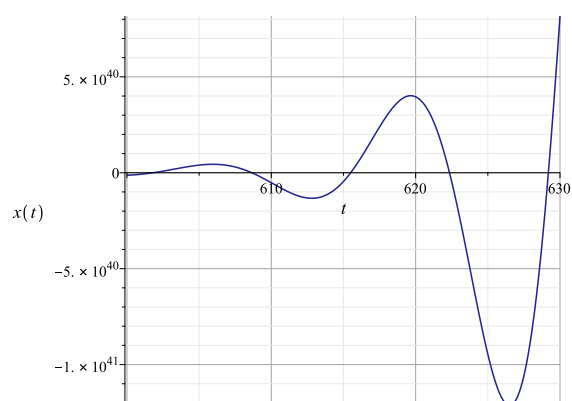


Рис. 4. Автоматизированный вывод численного решения задачи Коши (7) – (8) при параметре $L=[600, 630]$.

2.2. Сервисные программные процедуры автоматизированного построения 2d- и 3d-фазовых траекторий на основе численных решений

Автоматизированный вывод двумерной фазовой траектории численного решения методом **Method** задачи Коши **Eqs-ICS** на отрезке **L** в виде линии цвета **Color** реализуется программной процедурой **DifEqTools[GraphPhase2d](Eqs, ICS, Method, [[i1, j1], [i2, j2]], L, Color)**. При этом автоматически помечаются зеленой и красной точками начало и конец фазовой траектории.

```
> T:=time():
DifEqTools[GraphPhase2d](Eqs, ICS, [method=rkf45, maxfun=1000000], [[1, 0], [1, 1]],
32*Pi, COLOR(RGB, 0.5, 0, 0.5));
time()-T;

> T:=time():
DifEqTools[GraphPhase2d](Eqs, ICS, [method=rkf45], [[1, 0], [2, 1]], 32*Pi,
COLOR(RGB, 0, 0, 0.5));
time()-T;
```

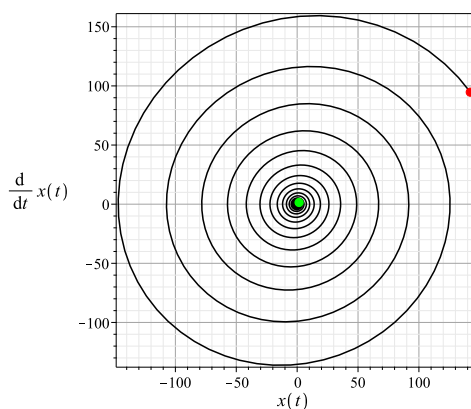


Рис. 5. Автоматизированный вывод фазовой траектории численного решения задачи Коши (5) – (6) $x(t), dx(t)/dt$ на отрезке $[0, 32\pi]$.

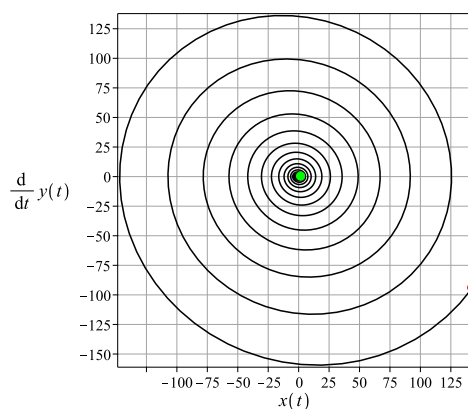


Рис. 6. Автоматизированный вывод фазовой траектории численного решения задачи Коши (5) – (6) $x(t), dy(t)/dt$ на отрезке $[0, 32\pi]$.

Автоматизированный вывод трехмерной фазовой траектории численного решения методом **Method** задачи Коши **Eqs-ICS** на отрезке **L** в виде линии цвета **Color** реализуется программной процедурой **DifEqTools[GraphPhase3d](Eqs,ICS,Method,[[i1,j1],[i2,j2],[i3,j3]],L,Color)**, где в списке **[[i1,j1],[i2,j2],[i3,j3]]** первые индексы **i1,i2,i3** есть порядковые номера выводимых функций (соответственно номеру исходного уравнения), а вторые индексы **[j1,j2,j3]** — порядки их выводимых производных. При этом следует отметить, что размерность соответствующей системы ОДУ может быть и больше 3, в этом случае мы получаем трехмерную проекцию конфигурационной траектории. При этом начало и конец траектории автоматически помечаются зеленой и красной точками, соответственно, фазовой.

Приведем пример вывода 3-х мерной фазовой траектории линейной системы ОДУ 2-го порядка.

```
> Eqs3:=(D@@2)(x)(t)=-D(y)(t)-x(t),(D@@2)(y)(t)=D(x)(t)-y(t),(D@@2)(z)(t)=0);
```

```
> ICS3:=[x(0)=0,D(x)(0)=1,y(0)=0,D(y)(0)=1,z(0)=0,D(z)(0)=1];
```

$$\text{Eqs3} := \left[D^{(2)}(x)(t) = -D(y)(t) - x(t), D^{(2)}(y)(t) = D(x)(t) - y(t), D^{(2)}(z)(t) = 0 \right] \quad (9)$$

$$\text{ICS3} := [x(0) = 0, D(x)(0) = 1, y(0) = 0, D(y)(0) = 1, z(0) = 0, D(z)(0) = 1] \quad (10)$$

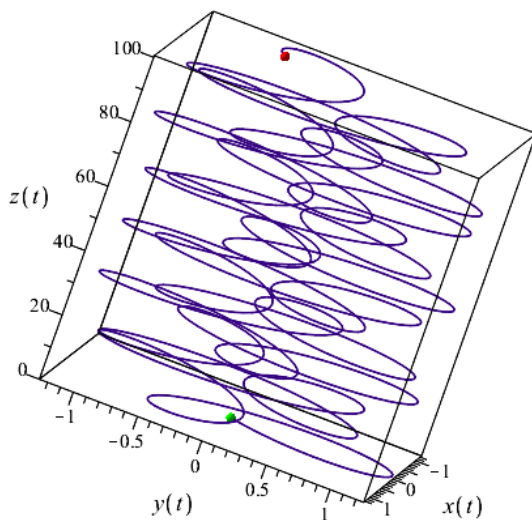


Рис. 7. Автоматизированный вывод фазовой траектории численного решения задачи Коши (9) — (10) методом **rkf45** в форме фазовой траектории: $[x(t), y(t), z(t)]$ на отрезке $[0, 32\pi]$. Время выполнения операции 0,32 сек.

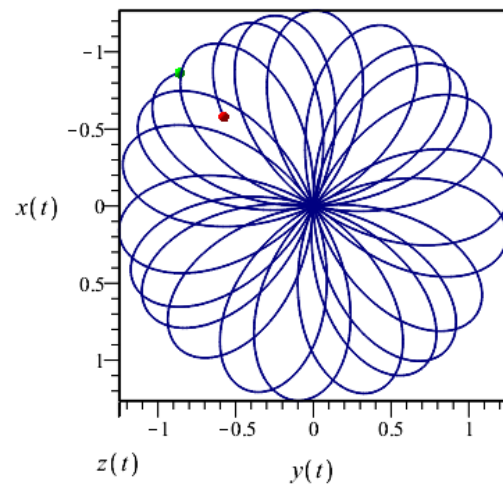


Рис. 8. Проекция фазовой траектории Рис. 7 на плоскость $\{XOY\}$ (углы Эйлера: $\text{orientation} = [\theta = 0, \psi = 0, \varphi = 0]$). Зеленая точка — начало траектории, красная — ее конец. Время построения траектории 0,33 сек.

```
> T:=time():
DifEqTools[GraphPhase3d](Eqs3,ICS3,[method=rkf45],[[1,0],[2,0],[3,0]],
32*Pi,COLOR(RGB,0.2,0,0.5));
time()-T;
```

Выше мы отмечали, что описанные программные процедуры с одинаковым успехом получают и реализуют численные решения, как линейных, так и нелинейных систем ОДУ. Продемонстрируем это на примере *существенно нелинейной системы ОДУ*, полученную из (9) заменой линейных членов в правых частях уравнений на кубические:

```
> Eqs3n:=(D@@2)(x)(t)=-D(y)(t)-x(t)^3,(D@@2)(y)(t)=D(x)(t)-y(t)^3,(D@@2)(z)(t)=0);
```

$$\left[D^{(2)}(x)(t) = -D(y)(t) - x(t)^3, D^{(2)}(y)(t) = D(x)(t) - y(t)^3, D^{(2)}(z)(t) = 0 \right] \quad (11)$$

```
> T:=time();
DifEqTools[GraphPhase3d](Eqs3n,ICS3,[method=rkf45],[[1,0],[2,0],[3,0]],
32*Pi,COLOR(RGB,0.2,0,0.5));
time()-T;
```

Видно, что несмотря на существенную нелинейность системы ОДУ (11) по сравнению с линейной системой (9) построение ее фазовой траектории по численным решениям заняло не более 10% дополнительного времени. При этом стоит подчеркнуть и существенное усложнение фазовой траектории — появившуюся ее угловатость.

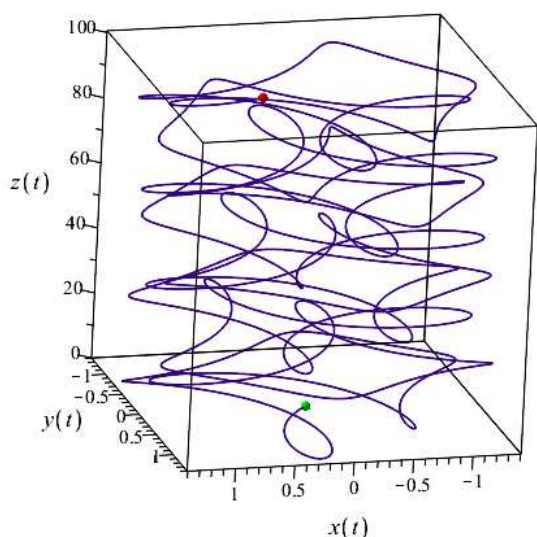


Рис. 9. Автоматизированный вывод численного решения задачи Коши (11), (10) методом `rkf45` в форме фазовой траектории: $[x(t), y(t), z(t)]$ на отрезке $[0, 32\pi]$. Время выполнения операции 0,35 сек.

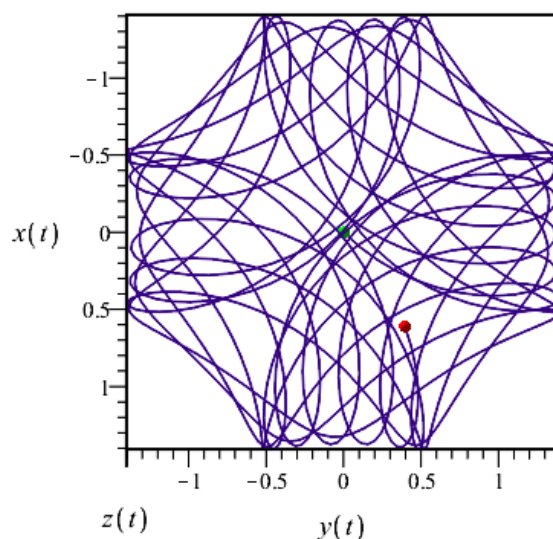


Рис. 10. Проекция фазовой траектории Рис. 9 на плоскость $\{XOY\}$ (углы Эйлера: $\text{orientation} = [\theta = 0, \psi = 0, \varphi = 0]$). Зеленая точка — начало траектории, красная — ее конец. Время построения траектории 0,35 сек.

2.3. Вывод графиков функций численных решений

Эти команды устроены аналогично предыдущим командам с той лишь разницей, что вводится последний список, `ff`, содержащий определения выводимых функций, а также, возможно, и их пользовательские имена: `DifEqTools[GraphFun](EQs,ICS,Method,L,ff,Color)`. В случае команды `GraphFun` этот список может состоять из двух или трех элементов. Первые два элемента задают, соответственно, функцию динамических переменных, а второй содержит масштабирование временной переменной (в случае, если масштабировать временную не требуется, необходимо положить второй элемент равным временной переменной). Третий, необязательный, элемент дает пользовательское имя выводимой по оси ординат функции. Если третий элемент отсутствует, на ось ординат выводится имя введенной функции. Кроме того, после всех обязательных параметров в опции команды `GraphFun` можно ввести число точек, по которым строится график.

Приведем примеры применения программной процедуры на примере численного интегрирования системы 4-х ОДУ 1 — 3 порядков.

```
> EQs:=[(D@@3)(x)(t)=-0.1*D(y)(t)-x(t),(D@@2)(y)(t)=-0.1*D(x)(t)-y(t),
  (D@@2)(z)(t)=-0.1*D(z)(t)-y(t),D(u)(t)=-0.1*D(z)(t)-x(t)];

> ICS:=[x(0)=2,D(x)(0)=1,(D@@2)(x)(0)=1/2,y(0)=0,D(y)(0)=0,z(0)=0,D(z)(0)=1,u(0)=0];
```

$$\begin{aligned} \text{EQs} := & [D^{(3)}(x)(t) = -0.1D(y)(t) - x(t), D^{(2)}(y)(t) = -0.1D(x)(t) - y(t), \\ & D^{(2)}(z)(t) = -0.1D(z)(t) - y(t), D(u)(t) = -0.1D(z)(t) - x(t)] \end{aligned} \quad (12)$$

$$\begin{aligned} \text{ICS4} := & [x(0) = 2, D(x)(0) = 1, D^{(2)}(x)(0) = 1/2, y(0) = 0, \\ & D(y)(0) = 0, z(0) = 0, D(z)(0) = 1, u(0) = 0] \end{aligned} \quad (13)$$

```
> DifEqTools[GraphFun](EQs,ICS4,[method=rkf45,abserr=10^(-6),relerr=10^(-5),maxfun=3000],
  [Pi,5*Pi],[sqrt(x(t)^2)*sin(y(t)),ln(t),sqrt('&Delta;&Psi;','(xi))],navy,numpoints=2000);

> DifEqTools[GraphFun](EQs,ICS4,[method=rkf45,abserr=10^(-6),relerr=10^(-5),maxfun=3000],
  [Pi,5*Pi],[sqrt(x(t)^2)*sin(y(t)),ln(t)],navy,numpoints=2000);
```

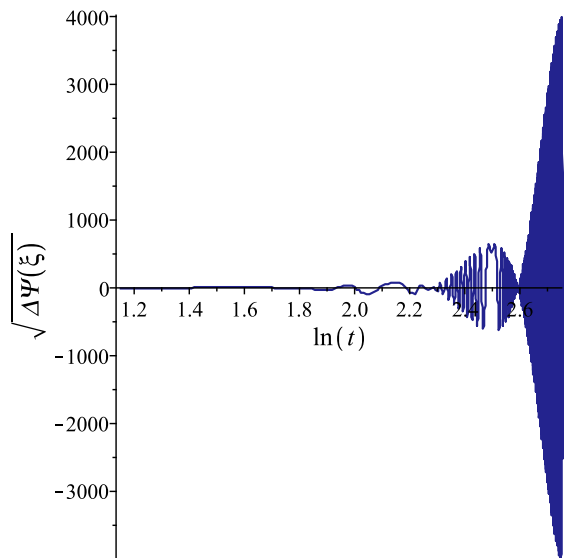


Рис. 11. Автоматизированный вывод численного решения задачи Коши (12), (6) методом rkf45 $[\ln t, \sqrt{x(t)^2 \sin(y(t))}]$ на отрезке $[\pi, 5\pi]$. На оси ординат выведено имя пользовательской функции $\sqrt{\Delta\psi(\xi)}$.

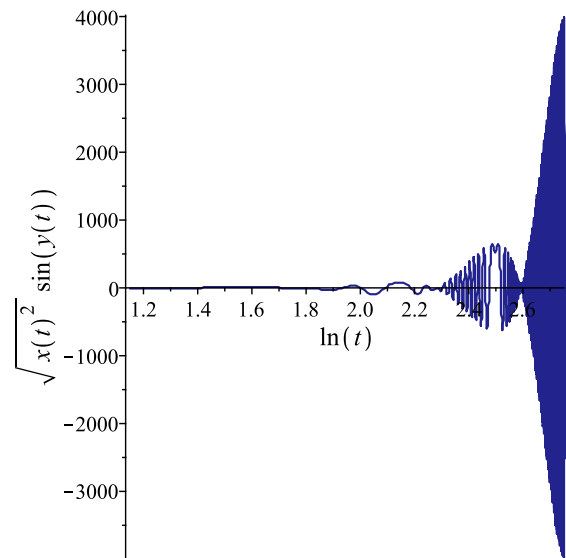


Рис. 12. Автоматизированный вывод численного решения задачи Коши (12), (6) методом rkf45 $[\ln t, \sqrt{x(t)^2 \sin(y(t))}]$ на отрезке $[\pi, 5\pi]$. На оси ординат выведено истинное имя функции $\sqrt{x(t)^2 \sin(y(t))}$.

Аналогично устроены и команды `GraphPhase2dFun` и `GraphPhase3dFun`. В этих командах построения фазовой траектории по двум координатным или трем осям выводятся функции динамических переменных. Возможностями этих команд можно воспользоваться, например, для построения фазовых траекторий динамических систем в криволинейных системах координат.

Продemonстрируем примеры построения 2-х и 3-х мерных фазовых траекторий уже рассмотренной выше динамической системы (12) – (6) в полярной

$$X = x(t) \cos(y(t)), \quad Y = x(t) \sin(y(t))$$

```
> DifEqTools[GraphPhase2dFun](EQs,ICS,[method=rkf45,abserr=10^(-6),relerr=10^(-5),
  maxfun=3000],[Pi,3*Pi],[[x(t)*cos(y(t)),x(t)*sin(y(t))]],navy,numpoints=20);
```


системе координат с числом точек `numpoints=20` (необязательный последний параметр) и цилиндрической системе координат:

$$X = x(t) \cos(y(t)), \quad Y = x(t) \sin(y(t)), \quad Z = z(t)$$

с числом точек `numpoints=5000` (необязательный последний параметр)

```
> DifEqTools[GraphPhase3dFun](EQs, ICS, [method=rkf45, abserr=10^(-6), relerr=10^(-5),
maxfun=5000], [0, 5*Pi], [[x(t)*sin(y(t)), x(t)*cos(y(t)), z(t)], [x, y, z]], navy,
numpoints=5000);
```

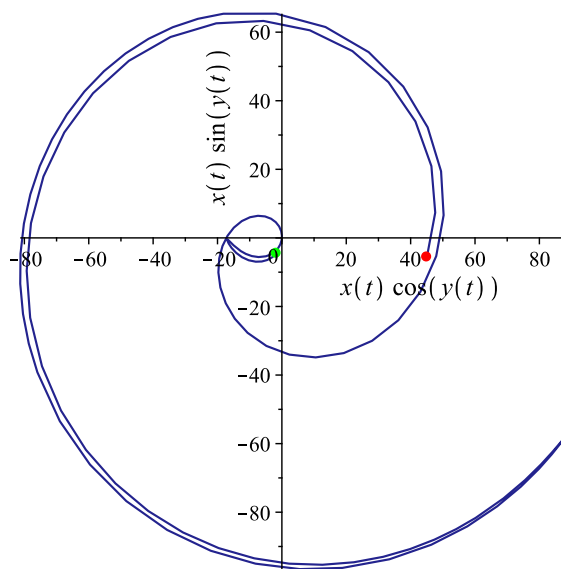


Рис. 13. Автоматизированный вывод численного решения задачи Коши (5) – (6) в форме двумерной фазовой траектории в полярной системе координат на отрезке $[\pi, 3\pi]$.

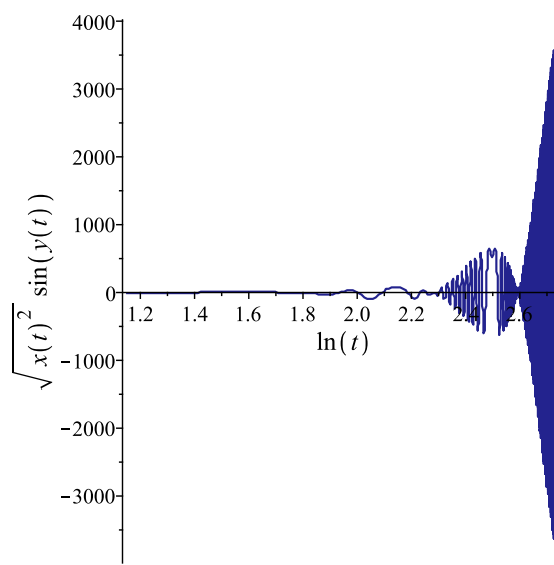


Рис. 14. Автоматизированный вывод численного решения задачи Коши (5) – (6) в форме трехмерной фазовой траектории в цилиндрической системе координат на отрезке $[0, 5\pi]$.

2.4. Сервисные программы 2d- и 3d- динамической визуализации нелинейных динамических систем

Пакет `DifEqTools` содержит программные процедуры автоматизированной динамической визуализации конфигурационных и фазовых траекторий нелинейных динамических систем:

- `DifEqTools[AnimConf](Eqs, ICS, Method, [i, j], L, Color, N);`
- `DifEqTools[AnimPhase2d](Eqs, ICS, Method, [i, j], L, Color, N);`
- `DifEqTools[AnimPhase2d](Eqs, ICS, Method, [[i1, j1], [i2, j2]], L, Color, N);`
- `DifEqTools[AnimPhase3d](Eqs, ICS, [method=rkf45], [[i1, j1], [i2, j2], [i3, j3]], L, Color, N);`
- `DifEqTools[AnimPhase2dFun](Eqs, ICS, Method, L, ff, Color, N);`
- `DifEqTools[AnimPhase3dFun](Eqs, ICS, Method, L, ff, Color, N);`

Эти программные процедуры отличаются от рассмотренных выше процедур статической визуализации лишь добавлением последнего целочисленного параметра `N` — числа кадров динамической модели. Необходимо отметить, что операция построения динамической компьютерной модели занимает гораздо больше времени, чем операция построения статической фазовой траектории хотя бы потому, что в процессе

построения динамической модели необходимо построить N статических 2d или 3d кадров. Кроме того, последним необязательным параметром этих процедур является `numpoints` — число точек, по которым строится каждый кадр анимации.

Приведем некоторые примеры создания динамических моделей.

Построим динамическую модель для системы (5)–(6), выводя временную зависимость первой производной второй неизвестной функции:

```
> T:=time():
DifEqTools[AnimConf](Eqs,ICS,[method=rkf45],[2,1],32*Pi,navy,32);
time()-T;
```

Также построим динамическую модель фазовой траектории для системы (5)–(6) в плоскости $\{x(t), \dot{x}(t)\}$:

```
> T:=time():
DifEqTools[AnimPhase2d](Eqs,ICS,[method=rkf45],[[1,0],[1,1]],32*Pi,
COLOR(RGB,0.5,0,0.5),32);
time()-T;
```

Таким образом, получим 2d-модели, состоящие из 32 кадров.

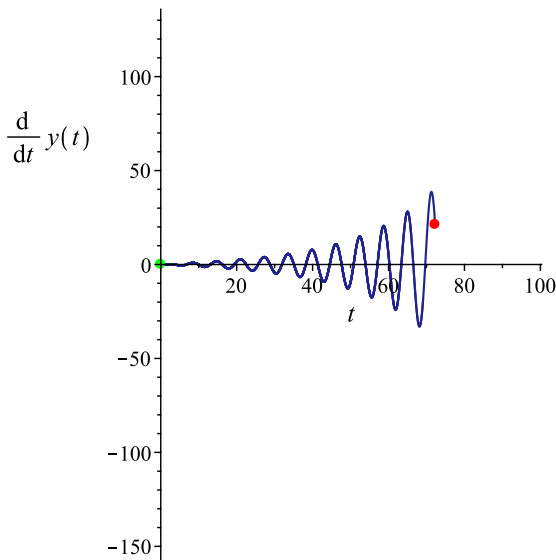


Рис. 15. 24-й кадр 32-х кадровый анимации численного решения задачи Коши (5)–(6) в форме конфигурационной траектории $\dot{y}(t)$ на отрезке $[0, 32\pi]$. Время создания модели 24 сек.

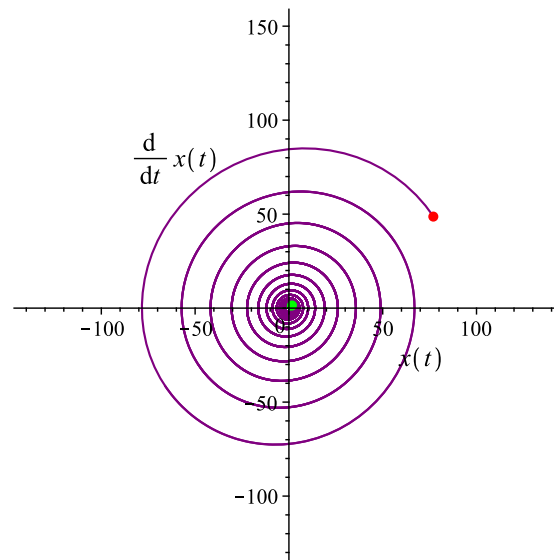


Рис. 16. 28-й кадр 32-х кадровый анимации численного решения задачи Коши (5)–(6) в форме фазовой траектории в плоскости $\{x(t), \dot{x}(t)\}$ на отрезке $[0, 32\pi]$. Время создания модели 50 сек.

Приведем примеры создания 3d - динамических моделей на примере системы (9)–(10) и заодно покажем пользу применения необязательного параметра.

В этом примере 8-й параметр опущен, число точек, `numpoints`, по которым будет построен график, по умолчанию равно 1000.

```
> T:=time():
> DifEqTools[AnimPhase3d](Eqs3,ICS3,[method=rkf45],[[1,0],[2,0],[3,0]],12*Pi,
COLOR(RGB,0.5,0,0.5),32);
time()-T;
```

В этом примере мы использовали необязательный 8-й параметр, положив `numpoints=300`.

```
> T:=time():
> DifEqTools[AnimPhase3d](Eqs3,ICS3,[method=rkf45],[[1,0],[2,0],[3,0]],12*Pi,
  COLOR(RED,0,0,0.5),32,300);
time()-T;
```

Таким образом, при сокращении числа точек графика в 3,3 раза скорость создания анимационной модели увеличивается в 3 раза. При этом не происходит заметного ухудшения качества графики. Этот наглядный пример следует иметь ввиду при построении конкретных динамических моделей. Приведенные примеры показывают необходимость оптимизации графической информации при построении анимационных моделей динамических систем, особенно, 3d-моделей.

Одной из возможностей такой оптимизации является применение сплайновой интерполяции полученных численных решений системы ОДУ. При этом сплайн можно сохранить во временной памяти и извлечь его при построении графика, а параметр `numpoints` можно увязать с числом узлов сплайна.

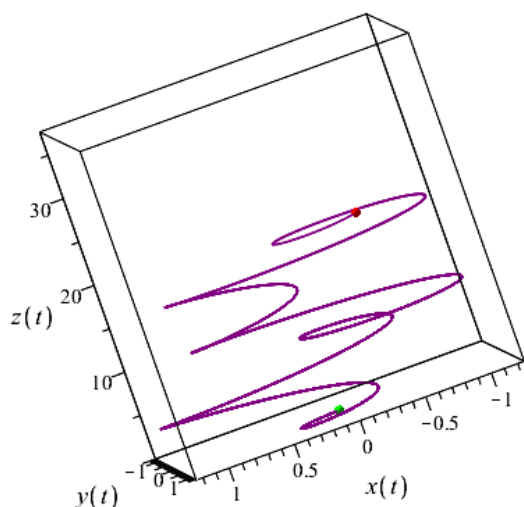


Рис. 17. 20-й кадр 32-х кадровой 3d-анимации численного решения задачи Коши (9)–(10) на отрезке $[0, 12\pi]$. Время создания модели 106 сек.

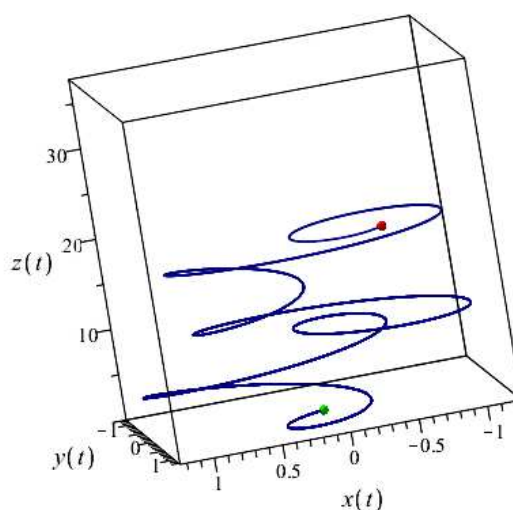


Рис. 18. 20-й кадр 32-х кадровой 3d-анимации численного решения задачи Коши (9)–(10) на отрезке $[0, 12\pi]$. Использован необязательный параметр `numpoints=300`. Время создания модели 35 сек.

3. Сервисные программные процедуры визуализации динамических моделей на основе сплайновой аппроксимации численных решений

3.1. Автоматизированный вывод численных решений системы ОДУ в форме сплайнов

Пакет `DifEqTools` содержит программную процедуру `ODESpline(Eqs,ICS,Method,t1,N,[i,j],x,r)` автоматизированного вывода численных решений системы ОДУ в форме равномерных N – кусочных сплайнов порядка r на отрезке $t1$, где x – пользовательское имя независимой переменной. Остальные обозначения совпадают с введенными ранее.

Приведем пример автоматизированного построения 12-ти кусочного кубического сплайна численного решения системы ОДУ (5) – (6) на отрезке $[t_0, 12\pi]$ относительно первой производной первой переменной с передачей имени x независимой (временной) переменной, причем для сокращения записи выведем сплайн с точностью до 4-х значащих цифр:

```
T:=time():
evalf(DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,12,[1,1],x,3),4);
time()-T;
```

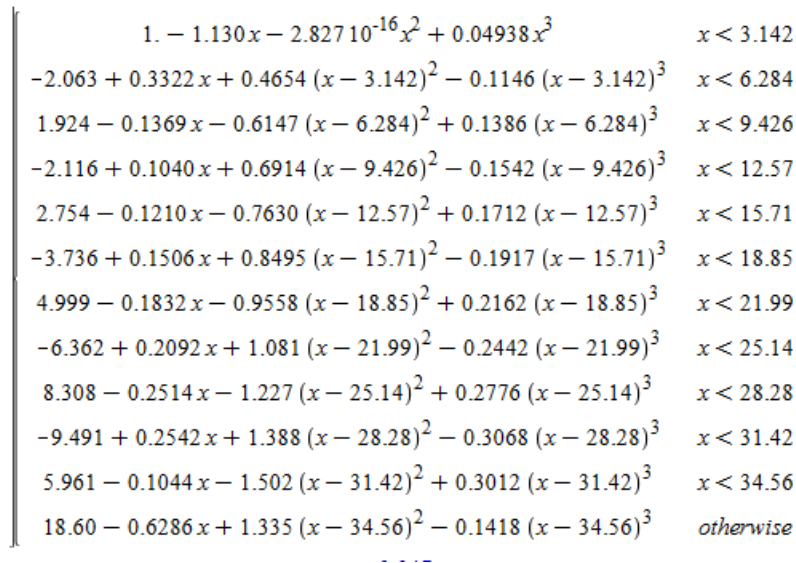


Рис. 19. 12-ти кусочный сплайн системы (5) – (6) на отрезке $[t_0, 12\pi]$. Время построения сплайна 0,03 сек.

Видно, что на построение 12-кусочного сплайна затрачено всего 0,03 сек. Но на построение 96-кусочного кубического сплайна затрачивается всего 0,09 сек!

```
> T:=time():
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,96,[1,1],x,3):
time()-T;
```

0.093

Поэтому для вывода решений можно не особенно экономить на числе кусков сплайна, а стараться обеспечивать точность решения системы ОДУ, которая, в первую очередь, определяется количеством кусков сплайна. Из приведенных ниже примеров видно, что 12-ти кусков явно недостаточно для обеспечения точности решения в сплайновом формате, а 96-ти кусков вполне хватает.

```
> DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,12,[1,0],10,3);
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,24,[1,0],10,3);
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,96,[1,0],10,3);
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,192,[1,0],10,3);
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45],12*Pi,384,[1,0],10,3);
```

```
-1.7170809520135988
-2.423982349087319
-2.481848771582304
-2.4820056339529297
-2.48200615724082
```

Заметим, еще раз, что основное время вычислений затрачивается не на численное решение дифференциальных уравнений и построение на основе этих решений сплайнов, а на выполнение графических процедур: ниже приведены два примера построения графиков численных решений по 48- и 12- кусочным сплайнам. Видно, что разница во времени построения графиков незначительна⁴.

```
> T:=time():
```

⁴В ряде примеров для экономии места мы не приводим в явном виде графики.

```
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45,abserr=10^(-6),relerr=10^(-7),
maxfun=500000],24*Pi,48,[1,1],x,3):
time()-T;
0.078
```

```
> T:=time():
DifEqTools[ODESpline](Eqs,ICS,[method=rkf45,abserr=10^(-8),relerr=10^(-8),
maxfun=200000],12*Pi,12,[1,1],x,3):
time()-T;
0.063
```

3.2. Автоматизированное построение конфигурационных и фазовых диаграмм на основе сплайновых решений

Нелинейные колебания в потенциальном поле 4-го порядка (этот пример будет подробно рассмотрен ниже):

```
> Eq_non:=[diff(x(t),t$2) = -.1*diff(x(t),t)+.1*x(t)-2*x(t)^3];
> IC0:=[x(0)=0,D(x)(0)=1];
```

Построим график $x(t)$ решения этого нелинейного уравнения колебаний на 248-кусочном сплайне:

```
> T:=time():
DifEqTools[SplineConf](Eq_non,IC0,[method=rkf45,abserr=10^(-8),relerr=10^(-8),
maxfun=200000],48*Pi,248,[1,0],x,3,blue);
time()-T;
```

Рассмотрим также систему двух нелинейных ОДУ второго порядка:

```
> Eqs2X2:=[(D@@2)(x)(t)=-0.1*D(y)(t)+0.1*x(t)-0.0001*y(t)^2,
(D@@2)(y)(t)=-0.1*D(x)(t)+0.1*y(t)-0.0001*x(t)*y(t)];
```

с начальными условиями

```
> ICS:=[x(0)=0,D(x)(0)=0.5,y(0)=0,D(y)(0)=0];
```

Построим график $y(t)$ этого нелинейного уравнения колебаний на 200-кусочном сплайне на отрезке $[0, 16\pi]$:

```
> T:=time():
DifEqTools[SplineConf](Eqs2X2,ICS,[method=rkf45],16*Pi,200,[2,0],x,3,navy);
time()-T;
```

Попытка проследить дальнейшую историю системы Eqs2X2 - ICS наталкивается на сообщение:

```
> T:=time():
DifEqTools[SplineConf](Eqs2X2,ICS,[method=rkf45],24*Pi,200,[2,0],x,3,blue);
time()-T;
```

Error, (in unknown) cannot evaluate the solution further right of 60.720201, maxfun limit exceeded (see ?dsolve,maxfun for details)

Система просит увеличить значение параметра численного интегрирования **maxfun**, равного максимальному числу шагов интегрирования. Эта проблема преодолевается улучшенными параметрами численного интегрирования, в частности, увеличением параметра **maxfun** до значения 5 000 000:

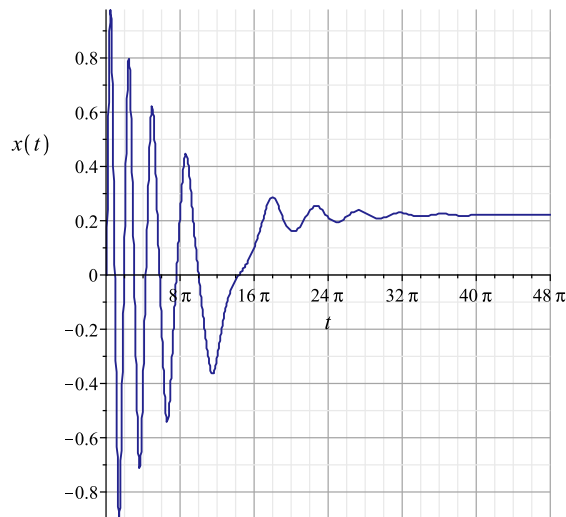


Рис. 20. Конфигурационная траектория численного решения задачи Коши $\text{Eq_non} - \text{IC0}$ на отрезке $[0, 48\pi]$, полученная на основе 248-кусочного сплайна. Время построения графика 0,31 сек. Надписи на координатных осях генерируются автоматически.

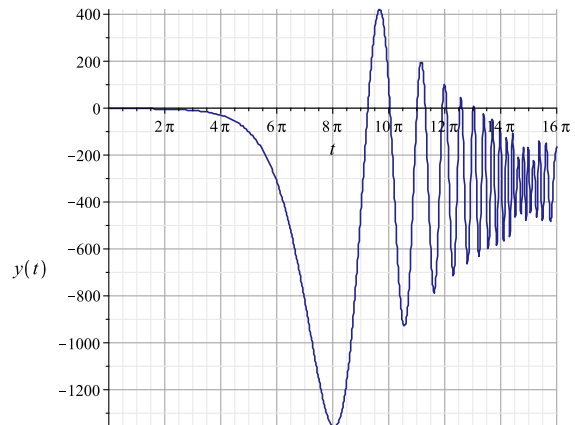


Рис. 21. Конфигурационная траектория численного решения задачи Коши $\text{Eqs2X2} - \text{ICS}$ на отрезке $[0, 16\pi]$, полученная на основе 200-кусочного сплайна. Время построения графика 0,25 сек. Надписи на координатных осях генерируются автоматически.

```
> T:=time():
DifEqTools[SplineConf](Eqs2X2,ICS,[method=rkf45,abserr=10^(-8),relerr=10^(-8),
maxfun=5000000],24*Pi,1000,[2,0],x,3,navy);
time()-T;
```

Рассмотрим также построение двумерной фазовой траектории для приведенного выше примера:

```
> T:=time():
DifEqTools[SplinePhase2d](Eq_non,IC0,[method=rkf45],48*Pi,248,[[1,0],[1,1]],x,3,navy);
time()-T;
```

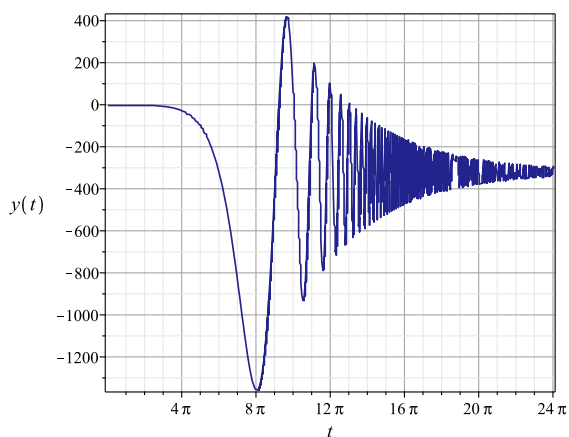


Рис. 22. Конфигурационная траектория численного решения задачи Коши $\text{Eqs2X2} - \text{ICS}$ на отрезке $[0, 24\pi]$, полученная на основе 1000 - кусочного сплайна с улучшенными опциями численного интегрирования. Время построения графика 2,9 сек.

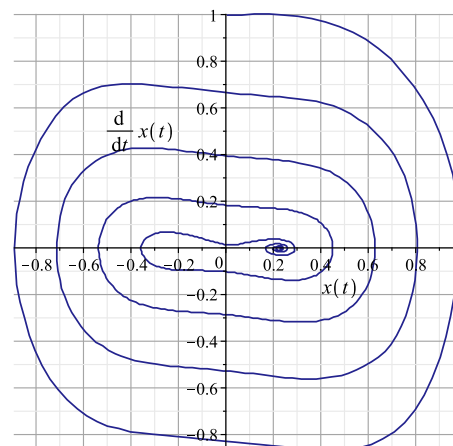


Рис. 23. Конфигурационная траектория численного решения задачи Коши $\text{Eq_non} - \text{IC0}$ на отрезке $[0, 48\pi]$, полученная на основе 248 - кусочного сплайна. Время построения графика 0,3 сек.

Приведем пример построения трехмерной фазовой траектории на основе сплайна для существенно нелинейной 3-х мерной динамической системы.

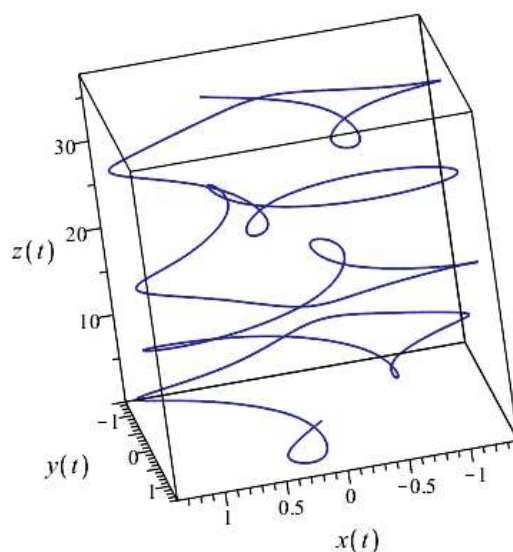


Рис. 24. Трехмерная фазовая траектория численного решения задачи Коши Eqs3n - ICS3 на отрезке $[0, 12\pi]$, полученная на основе 48 - кусочного кубического сплайна. Время построения графика 0,3 сек.

```
> Eqs3n:=[(D@@2)(x)(t)=-D(y)(t)-x(t)^3,(D@@2)(y)(t)=D(x)(t)-y(t)^3,(D@@2)(z)(t)=0];

> ICS3:=[x(0)=0,D(x)(0)=1,y(0)=0,D(y)(0)=1,z(0)=0,D(z)(0)=1];
      [x(0) = 0, D(x)(0) = 1, y(0) = 0, D(y)(0) = 1, z(0) = 0,
      D(z)(0) = 1];

> DifEqTools[SplinePhase3d](Eqs3n,ICS3,
      [method=rkf45,abserr=10^(-8),relerr=10^(-8),maxfun=5000000],
      12*Pi,48,[[1,0],[2,0],[3,0]],x,3,navy);
time()-T;
```

Отметим, что на основе созданных процедур построения фазовых траекторий можно строить фазовые траектории (вернее, их проекции) и для 4-х мерных и n-мерных. Рассмотрим пример 4-х мерной нелинейной системы:

```
> Eqs4n:=[(D@@2)(x)(t)=-D(y)(t)-x(t)^3+0.01*u(t)^3,
      (D@@2)(y)(t)=D(x)(t)-y(t)^3+0.01*z(t)^3,(D@@2)(z)(t)=0.01*u(t),
      (D@@2)(u)(t)=-D(z)(t)-0.1*y(t)^3];

> ICS4:=[x(0)=0,D(x)(0)=1,y(0)=0,D(y)(0)=1,z(0)=0,D(z)(0)=1,u(0)=1,D(u)(0)=0];
```

Следующие два примера показывают, как может возникнуть мнимая «сингулярность» в точке $t=25.787255$:

```
> T:=time();
DifEqTools[SplinePhase3d](Eqs4n,ICS4,[method=rkf45],26,96,[[1,0],[2,0],[4,0]],x,3,navy);
time()-T;
```

Error, (in unknown) cannot evaluate the solution further right of 25.787255, maxfun limit exceeded (see ?dsolve,maxfun for details)

Улучшим параметры численного интегрирования.

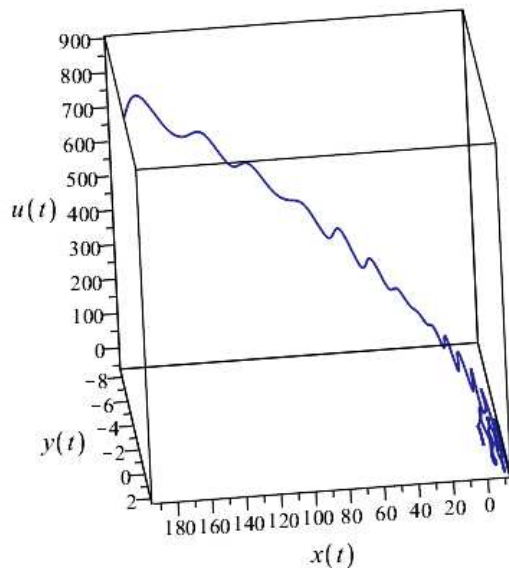


Рис. 25. Трехмерная проекция в гиперплоскости $\{XYU\}$ четырехмерной фазовой траектории численного решения задачи Коши Eqs4n - ICS4 на отрезке $[0, 26]$, полученная на основе 96 - кусочного сплайна. Время построения графика 0,9 сек.

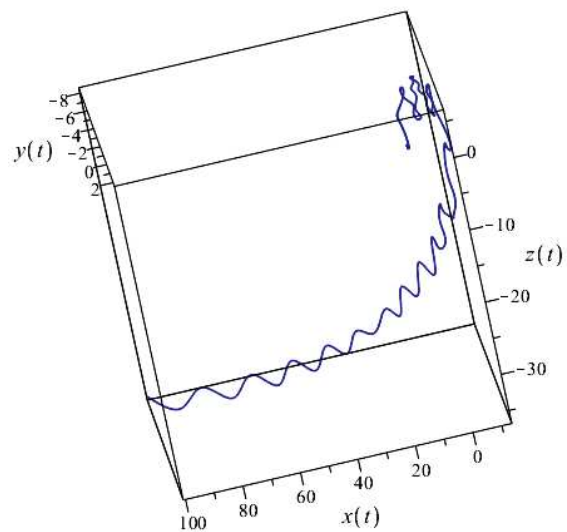


Рис. 26. Трехмерная проекция в гиперплоскости $\{XYZ\}$ четырехмерной фазовой траектории численного решения задачи Коши Eqs4n - ICS4 на отрезке $[0, 24]$, полученная на основе 100 - кусочного сплайна. Время построения графика 0,8 сек.

```
> T:=time():
DifEqTools[SplinePhase3d](Eqs4n,ICS4,[method=rkf45,abserr=10^(-6),relerr=10^(-6),
maxfun=10000000],26,96,[[1,0],[2,0],[4,0]],x,3,navy);
time()-T;

> T:=time():
DifEqTools[SplinePhase3d](Eqs4n,ICS4,[method=rkf45],24,100,
[[1,0],[2,0],[3,0]],x,3,navy);
time()-T;
```

4. Программные процедуры автоматизированного усреднения функций численных решений

Часто для анализа поведения нелинейных динамических систем необходимо выделить медленно меняющуюся и быстро осциллирующую части этой функции:

$$F(t) = f(t) + \phi(t)e^{iU(t)}, \quad |\dot{f}, \dot{\phi}| \ll |\dot{U}|. \quad (14)$$

Пусть $F(t)$ — некоторая интегрируемая функция. Определим ее среднее значение $F_0(t)$ на промежутке $[t - \Delta t/2, t + \Delta t/2]$ следующим образом:

$$F_0(t) = \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} F(t') dt'. \quad (15)$$

Также определим на этом же промежутке флуктуации функции $\delta F(t)$:

$$\delta F(t) = F(t) - F_0(t). \quad (16)$$

Также определим среднеквадратичные флуктуации:

$$\langle \delta F(t)^2 \rangle = \frac{1}{\Delta t} \int_{t-\Delta t/2}^{t+\Delta t/2} \delta F^2(t') dt'. \quad (17)$$

Заметим, что для уверенного отделения осциллирующей части функции необходимо выполнение 2-х условий:

$$T \ll \Delta t \ll t, \quad (18)$$

где T — характерный период осцилляций. Программные процедуры, соответствующие определениям (15) — (17), в пакете `DifEqTools` реализуются следующим образом:

1. `DifEqTools[Middle](ff,T,tau,N,k)` — (15);
2. `DifEqTools[Osc](ff,T,tau,N,k)` — (16);
3. `DifEqTools[OscQuadr](ff,T,tau,N,k)` — (17),

где $ff=[t, F(t)]$, $T = \Delta t$, τ — независимая временная переменная в левых частях формул (15) — (17), N — число разбиений интервала усреднения $T = \Delta t$, k — метод интегрирования: $k=T$ — метод трапеций, $k=S$ — метод Симпсона.

В качестве примера рассмотрим следующую функцию:

```
> phi:=(t)->1/(1+t^2)+\frac{0.1*\sin(10*t)}{(1+t^2)};
```

$$\phi := t \rightarrow \frac{1}{1+t^2} + 0.1 \frac{\sin(10t)}{1+t^2} \quad (19)$$

Определим среднее этой функции по интервалу $\pi/5$, заметив, что этот интервал соответствует одному периоду колебаний. Вычислим среднее значение с помощью метода интегрирования Симпсона, разбивая отрезок интегрирования на 1000 частей:

```
> phiS_0:=(tau)->DifEqTools[Middle]([t,phi(t)],Pi/5,tau,1000,S);
```

На рисунках 27, 28 показаны графики функции $\phi(\tau)$ и ее среднего значения.

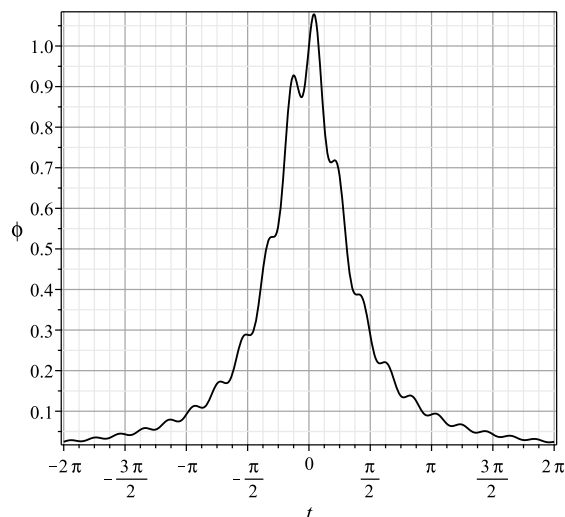


Рис. 27. График функции $\phi(\tau)$ (19).

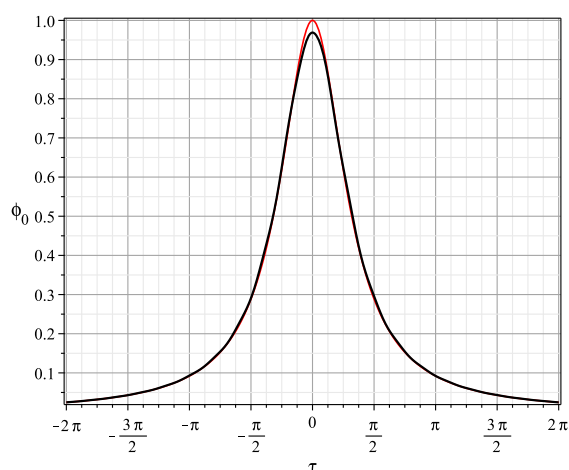


Рис. 28. График среднего значения функции $\phi(\tau)$, — черная жирная линия; красная тонкая линия — график медленно меняющейся части этой функции: $1/(1+\tau^2)$.

Выделим теперь осциллирующую часть функции $\phi(t)$ с помощью команды и построим график осцилляций (Рис. 29):

```
> plot(DifEqTools[Osc]([t,phi(t)],Pi/5,tau,1000,S),tau=-2*Pi..2*Pi);
```

Построим также график среднеквадратичных флуктуаций этой функции (Рис. 30):

```
> d2phi:=(tau)->DifEqTools[OscQuadr]([t,phi(t)],Pi/50,tau,100,S);
> plot(d2phi(tau),tau=-2*Pi..2*Pi);
```

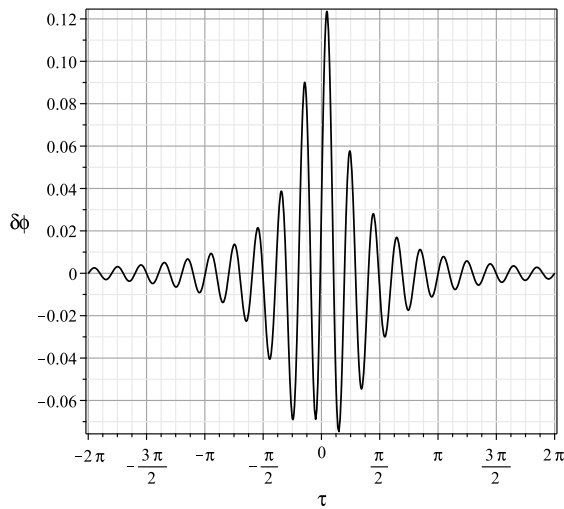


Рис. 29. Осцилляции функции $\phi(\tau)$.

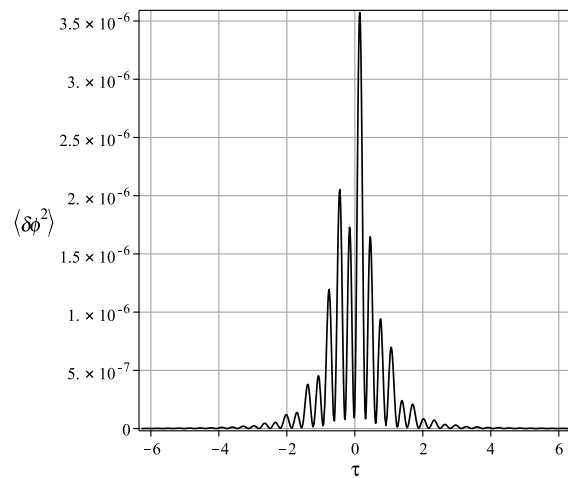


Рис. 30. Среднеквадратичные флуктуации функции $\phi(\tau)$.

Аналогично предыдущим определены и программные процедуры усреднения функций численных решений нелинейной системы обыкновенных дифференциальных уравнений:

1. `DifEqTools[NumericDsolveMiddle](Eqs,ICs,metod,ff,T,tau,N,k);`
2. `DifEqTools[NumericDsolveOsc](Eqs,ICs,metod,ff,T,tau,N,k);`
3. `DifEqTools[NumericDsolveQuadr](Eqs,ICs,metod,ff,T,tau,N,k),`

где `Eqs` — упорядоченная система ОДУ, `ICs` — упорядоченная система начальных условий, `metod` — упорядоченная система опций метода интегрирования ОДУ, остальные параметры команд такие же, как и для команд усреднения заданных функций.

Покажем реализацию команд на примере уравнения свободных колебаний.

```
> Eq_osc:=[diff(x(t),t$2)=-x(t)];
```

$$\frac{d^2}{dt^2}x(t) = -x(t) \quad (20)$$

с начальными условиями

```
> IC_osc:=[x(0)=0,D(x)(0)=1];
```

$$IC_{osc} := [x(0) = 0, D(x)(0) = 1] \quad (21)$$

Определим среднее на численном решении задачи Коши (20) – (21), заодно продемонстрировав возможность манипуляции параметрами опции `metod`.

```
> G:=(tau)->DifEqTools[NumericDsolveMiddle](Eq_osc,IC_osc,[method=rkf45,
maxfun=50000,abserr = 1.*10^(-7), relerr = 1.*10^(-7)], [t,sin(x(t))],2*Pi,tau,1000,S);
```

Увеличим точность решения ОДУ на порядок

```
> G1:=(tau)->DifEqTools[NumericDsolveMiddle](Eq_osc,IC_osc,[method=dverk78,
maxfun=50000,abserr = 1.*10^(-8), relerr = 1.*10^(-8)], [t,sin(x(t))],2*Pi,tau,1000,S);
```

и построим соответствующие графики, учитывая тот факт, что точным решением указанной задачи Коши является $x(t) = \sin(t)$.

```
> GM1:=plot(G(tau),tau=0..4*Pi,labels=[tau,phi__0],label font=[TIMES,ROMAN,14],
  color=black);
  plot(G1(tau),tau=0..4*Pi,labels=[tau,phi__0],label font=[TIMES,ROMAN,14]);
```

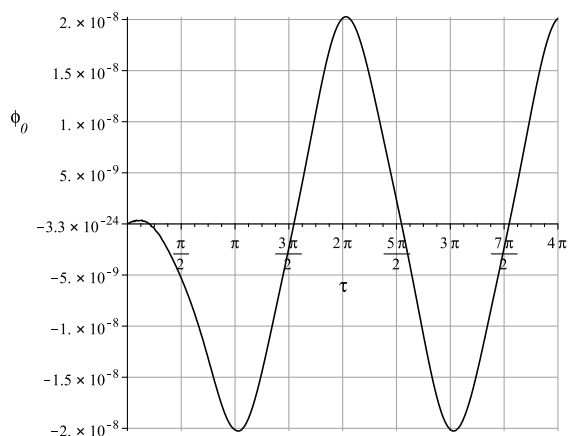


Рис. 31. Среднее значение численного решения уравнения свободных колебаний, вычисленного с точностью 10^{-7} .

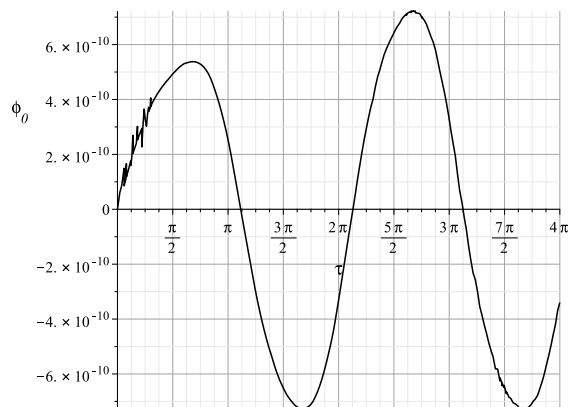


Рис. 32. Среднее значение численного решения уравнения свободных колебаний, вычисленного с точностью 10^{-8} .

Поскольку решение является строго периодической (и даже гармонической) функцией, среднее его значение должно равняться нулю. Приведенные на рисунках 31 – 32 графики как раз и показывают это. При этом увеличение абсолютной и относительной точности численного решения ОДУ на 1 порядок дает почти на 2 порядка лучший результат.

Вследствие равенства нулю среднего значения функции ее флуктуации должны совпадать со значениями самой функции. Проверим это.

```
> GSin1:=plot(sin(sin(t)),t=0..3*Pi,color=black,linestyle=dashdot);
  plots[display](GSin1,0sGM1);
```

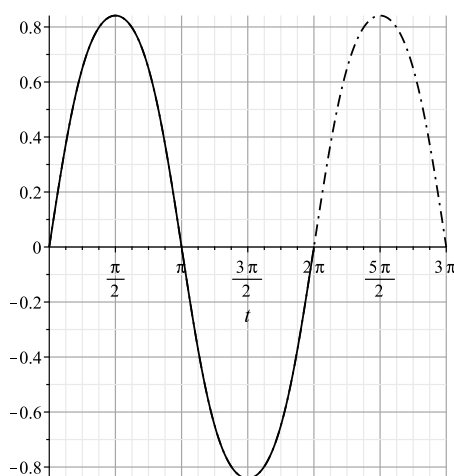


Рис. 33. Совмещение графиков осцилляции численного решения (сплошная линия) и графика $\sin t$ – пунктирная линия.

Мы видим полное совпадение графиков.

5. Приложения к нелинейным механическим системам

Как известно, колебания связанных систем вблизи точки равновесия являются одним из наиболее фундаментальных их свойств. Присущи такие колебания и нелинейным динамическим системам. Поэтому для анализа таких динамических систем важно выделить быстро осциллирующую и медленно меняющуюся части. Аналитически такая задача решается методом Боголюбова – Крылова [9]. Обычно осциллирующие части динамических переменных связывают с кинетической энергией колебаний, а медленно меняющиеся части — с внутренней энергией системы. Этому подходу соответствует и квантово – механическая / полевая картина, в которой колебания связываются с квазичастицами, а медленно меняющаяся часть — с вакуумом. Созданные нами программные процедуры позволяют проводить соответствующие операции усреднения и выделения быстрых осцилляций.

5.1. Нелинейные одномерные колебания в потенциальном поле Хиггса и Хиггсов механизм спонтанного нарушения симметрии

Важнейшим примером нелинейной механической системы является модель одномерных механических колебаний в поле потенциала Хиггса, представляющем потенциал 4-го порядка (см., например, [5], [6]). Соответствующее уравнение колебаний имеет вид⁵:

$$\frac{d^2x}{dt^2} + \beta \frac{dx}{dt} - x + x^3 = 0, \quad (22)$$

где β — коэффициент трения. Потенциал Хиггса описывается симметричной потенциальной ямой с двумя минимумами и одним центральным максимумом, что позволяет описать простейший механизм Хиггса спонтанного нарушения симметрии. Последнее заключается в том, что в результате диссипации энергии свободные колебания системы со временем затухают, в результате чего система оказывается в одном из минимумов (левом или правом).

5.2. Фазовые траектории

Покажем, как с помощью простых команд можно реализовать конфигурационную и фазовую траектории нелинейных колебаний. Введем уравнение (22) с коэффициентом $\beta = 0.05$.

```
> EqNon:=(D@@2)(x)(t)=-0.05*D(x)(t)+x(t)-x(t)^3;
```

$$EqNon := [(D@@2)(x)(t) = -0.05D(x)(t) + x(t) - x(t)^3] \quad (23)$$

Введем теперь начальные условия задачи:

```
> IC:=[x(0)=0,D(x)(0)=5];
```

$$IC := [x(0) = 0, (D(x))(0) = 5] \quad (24)$$

и, наконец, с помощью команд `GraphConf1` и `GraphPhase2d` построим конфигурационную и фазовую траектории системы (Рис. 34, 35)⁶:

```
> DifEqTools[GraphConf1](EqNon,IC,[method=rkf45],[1,0],48*Pi,navy);
```

```
> T:=time();
```

```
> DifEqTools[GraphPhase2d](EqNon,IC,[method=rkf45,maxfun=1000000],
```

```
[[1,0],[1,1]],32*Pi,COLOR(RGB,0.0,0.0,0.5));
```

```
time()-T;
```

⁵Уравнение колебаний записано в безразмерных переменных, к которым можно перейти с помощью масштабных преобразований.

⁶Подключена функция `time` для автоматического вычисления времени исполнения команды.

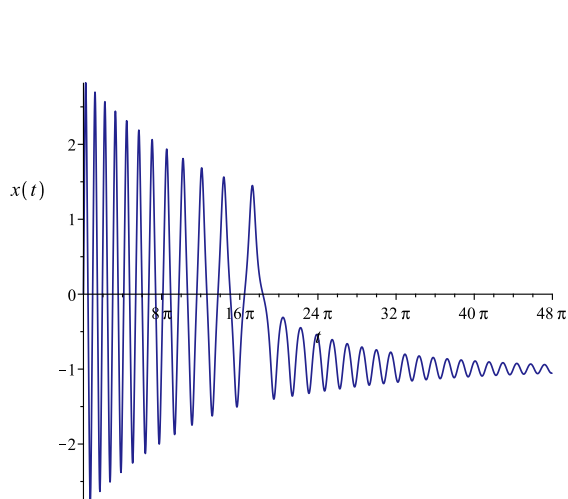


Рис. 34. Конфигурационная траектория динамической системы (23) с начальными условиями (24). Интегрирование проведено методом `rkf45`. Имена координатных осей сгенерированы автоматически. Нарушение симметрии налево происходит в момент времени $t \sim 16\pi$.

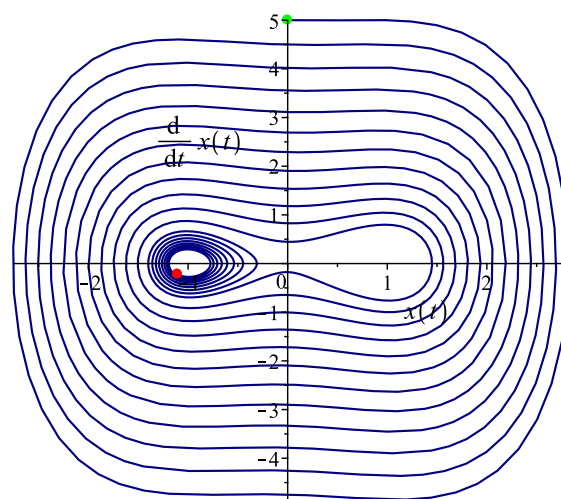


Рис. 35. Фазовая траектория динамической системы (23) с начальными условиями (24). Интегрирование проведено методом `rkf45` с опцией `maxfun=1000000`. Зеленой точкой автоматически помечается начало траектории, красной ее конец. Время исполнения $\Delta t = 0.234$ с.

5.3. Усреднение колебаний по быстрым осцилляциям

Приведенные на рисунках 34, 35 графики показывают возможность и необходимость исследования осцилляций нелинейных колебаний. Действительно, во-первых, графики обнаруживают колебательный характер движения, а, во-вторых, амплитуда этих колебаний падает достаточно медленно, кроме критического участка в области нарушения симметрии. Вычислим среднее по численному решению уравнения (23) с помощью описанной выше программной процедуры (см. Рис. 36 и 37).

```
> GN:=(tau)->DifEqTools[NumericDsolveMiddle](EqNon,IC,[method=dverk78,
maxfun=500000,abserr = 1.*10^(-6),
relerr = 1.*10^(-6)], [t,x(t)],Pi,tau,100,S);

> GN(1);GN(100);
-0.15527961939736498
-0.9713948576870406

> GN_middle:=plot(GN(tau),tau=0..48*Pi,labels=[t,x__0(t)],
labelfont=[TIMES,ROMAN,14],color=black);

> GN_Osc:=(tau)->DifEqTools[NumericDsolveOsc](EqNon,IC,[method=dverk78,maxfun=500000,
abserr = 1.*10^(-6), relerr = 1.*10^(-6)], [t,x(t)],Pi,tau,100,S);
```

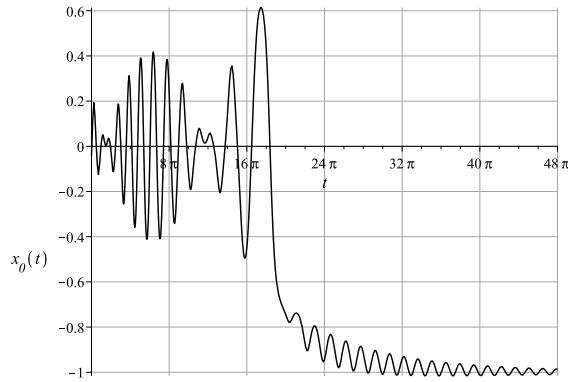


Рис. 36. Среднее от амплитуды колебаний задачи Коши (23) – (24). Усреднение проведено по длительности π .

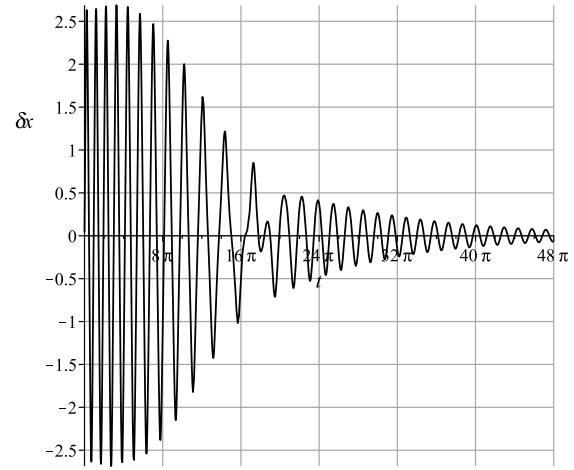


Рис. 37. Выделение осцилляций колебательного процесса, описываемого задачей Коши (23) – (24).

6. Приложения к космологии

6.1. Стандартная космологическая модель

Как известно, для пространственно-плоской космологической модели

$$ds^2 = dt^2 - a^2(t)(dx^2 + dy^2 + dz^2) \quad (25)$$

система уравнений Эйнштейна с космологическим членом λ и Клейна-Гордона принимает вид ($\dot{f} \equiv df/dt$):

$$\begin{aligned} 3\frac{\dot{a}^2}{a^2} &= \dot{\Phi}^2 + m^2\Phi^2 + \lambda \Rightarrow \\ H^2 \equiv \dot{\Lambda}^2 &= \frac{1}{3}(\dot{\Phi}^2 + m^2\Phi^2 + \lambda) \end{aligned} \quad (26)$$

— единственное нетривиальное уравнение Эйнштейна и уравнение классического массивного скалярного поля:

$$\ddot{\Phi} + 3H\dot{\Phi} + m^2\Phi = 0, \quad (27)$$

где $H(t)$ — постоянная Хаббла

$$\Lambda = \ln(a); \quad \frac{\dot{a}}{a} \equiv \dot{\Lambda} = H(t). \quad (28)$$

При этом тензор энергии — импульса скалярного поля имеет структуру тензора энергии — импульса изотропной жидкости с плотностью энергии и давлением:

$$\varepsilon = \frac{1}{8\pi}(\dot{\Phi}^2 + m^2\Phi^2); \quad p = \frac{1}{8\pi}(\dot{\Phi}^2 - m^2\Phi^2). \quad (29)$$

6.2. Приведение системы уравнений к нормальному виду

Пользуясь тем, что постоянную Хаббла можно выразить из уравнения Эйнштейна (26) через функции Φ , $\dot{\Phi}$, переходя к безразмерному комптоновскому времени:

$$mt = \tau; \quad (m \neq 0)$$

и проводя стандартную замену переменных $\Phi' = Z(t)$, приведем систему уравнений поля (26), (27) к безразмерному виду.

6.3. Космологические фазовые траектории скалярного поля

Итак, запишем уравнение поля (27) в виде:

```
> EqF:=(lambda)->[(D@@2)(Phi)(tau)=-sqrt(3*Pi)*sqrt(lambda+8*Pi*((D(Phi)(tau))^2+
Phi(tau)^2))*D(Phi)(tau)-Phi(tau)];
```

$$EqF := \lambda \rightarrow \left[D^{(2)}(\Phi)(\tau) = -\sqrt{3\pi} \sqrt{\lambda + 8\pi (D(\Phi)(\tau)^2 + \Phi(\tau)^2)} D(\Phi)(\tau) - \Phi(\tau) \right] \quad (30)$$

```
> convert(EqF(lambda),diff);
```

$$\left[\frac{d^2\Phi(\tau)}{d\tau^2} = -\sqrt{3\pi} \sqrt{\lambda + 8\pi \left[\left(\frac{d\Phi(\tau)}{d\tau} \right)^2 + \Phi(\tau)^2 \right]} \frac{d\Phi(\tau)}{d\tau} - \Phi(\tau) \right] \quad (31)$$

Зададим далее начальные условия

```
> ICF:=[Phi(0)=1,D(Phi)(0)=0];
```

$$ICF := [\Phi(0) = 1, D(\Phi)(0) = 0] \quad (32)$$

и выведем фазовую траекторию скалярного поля на большом временном интервале $t \in [0, 1000]$ при $\lambda = 0,001$ с помощью команды `DifEqTools[GraphPhase2d]`:

```
> DifEqTools[GraphPhase2d](EqF(0.001),ICF,[method=rkf45],
[[1,0],[1,1]],[0,1000],COLOR(RGB,0,0,0.5));
```

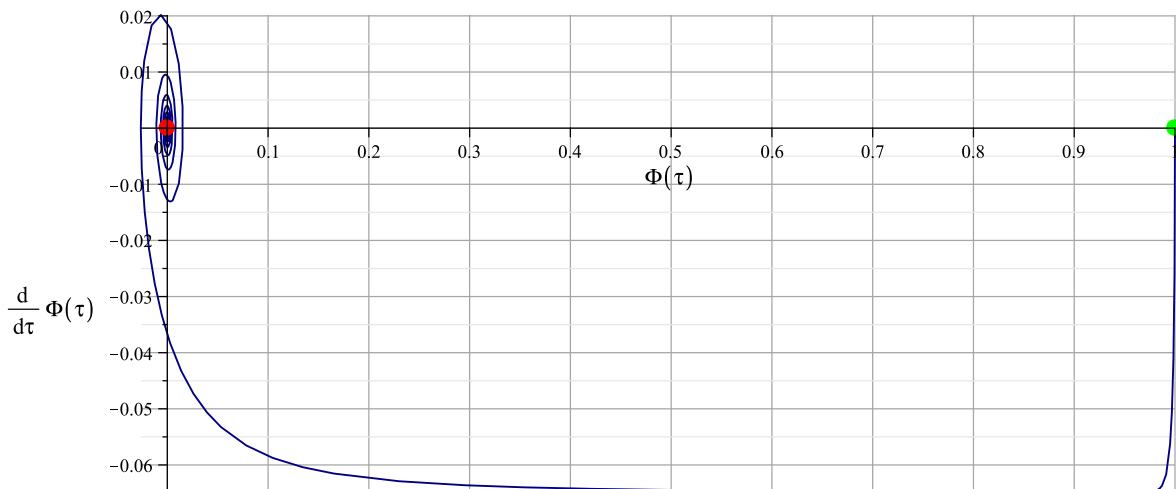


Рис. 38. Космологическая фазовая траектория классического скалярного поля на больших временных интервалах $\tau = 0 \div 10^3$. Зеленой точкой автоматически отмечено начало траектории, красной — ее конец; названия координатных осей генерируются автоматически.

Выведем теперь поздний участок эволюции скалярного поля на временном интервале $t \in [900, 2000]$ при $\lambda = 0,001$:

```
> T:=time();
DifEqTools[GraphPhase2d](EqF(0.001),ICF,[method=rkf45],[[1,0],[1,1]],
[900,2000],COLOR(RGB,0,0,0.5));
time()-T;
```

и на временном интервале $t \in [900, 1000]$ при $\lambda = 0,001$:


```
> T:=time():
DifEqTools[GraphPhase2d](EqF(0.001),ICF,[method=rkf45],[[1,0],[1,1]],
[900,1000],COLOR(RGB,0,0,0.5));
time()-T;
```

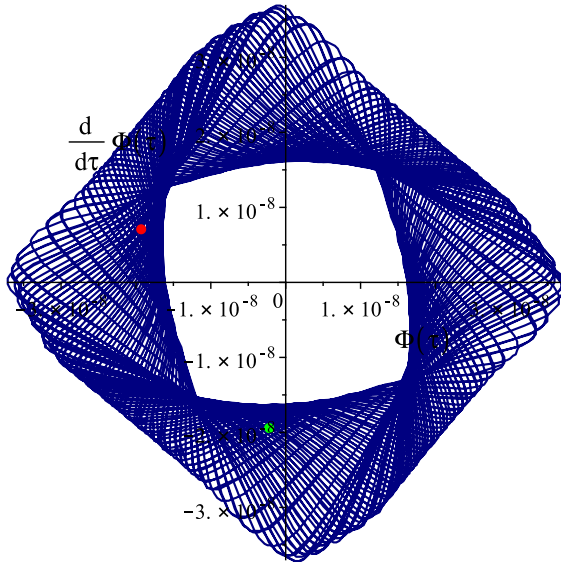


Рис. 39. Метод rkf45, время вычисления $\Delta t = 1,2$ сек.

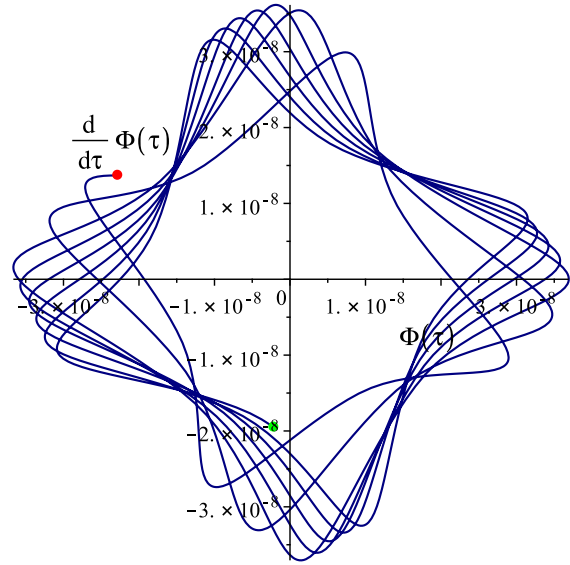


Рис. 40. Метод rkf45, время вычисления $\Delta t = 0.187$ сек.

Странное поведение решения на поздних временах эволюции свидетельствует о недостаточной точности метода численного интегрирования `rkf45`. Заменяв метод интегрирования на более точный — метод Рунге - Кутты 7-8 порядков, получим правильное решение [7]:

```
> T:=time():
DifEqTools[GraphPhase2d](EqF(0.001),ICF,[method=dverk78],[[1,0],[1,1]],
[9000,10000],COLOR(RGB,0,0,0.5));
time()-T;
```

```
> T:=time():
DifEqTools[GraphPhase2d](EqF(0),ICF,[method=dverk78],[[1,0],[1,1]],
[99900,100000],COLOR(RGB,0,0,0.5));
time()-T;
```

Заметим, что на поздних временах эволюции $\tau \gg 1$, во-первых,

$$|\Phi| \approx |\dot{\Phi}| \quad (33)$$

и, во-вторых, эти величины весьма медленно убывают. Это дало повод предположить, что при $\tau \rightarrow +\infty$ система имеет конечный цикл [8]. Для выяснения поведения системы на конечных стадиях эволюции и понадобился более точный численный анализ, который был осуществлен указанными выше средствами пакета `DifEqTools` [7].

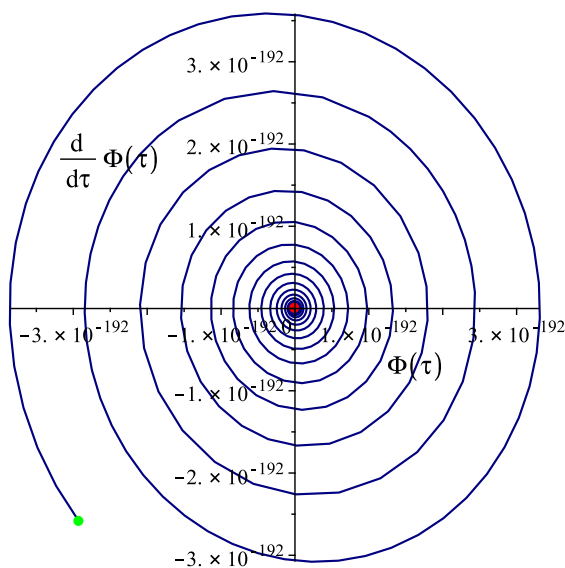


Рис. 41. Метод dverk78, время вычисления $\Delta t = 15.367$ сек.

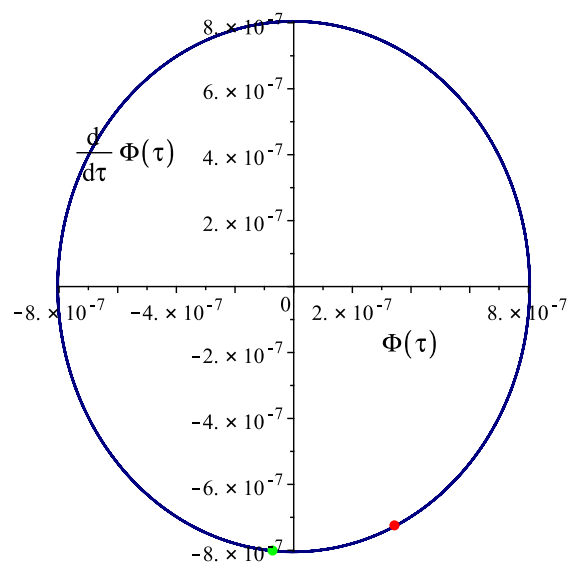


Рис. 42. Метод dverk78, время вычисления $\Delta t = 11.888$ сек.

6.4. Усреднение космологического скалярного поля по быстрым осцилляциям

Таким образом, космологическая модель, основанная на классическом скалярном поле, обнаруживает типичные черты нелинейной динамической системы со слабой диссипацией энергии: быстрые осцилляции, с одной стороны, и их медленное затухание — с другой. Это говорит о необходимости анализа средних величин и выделения быстро осциллирующей части. Именно с этой целью и были созданы описанные выше программные процедуры усреднения.

Вычислим сначала средние величины динамических переменных $\langle \Phi(t) \rangle \equiv \Phi_0(t)$ и $\langle Z(t) \rangle \equiv Z_0(t)$ с помощью программных процедур усреднения `DifEqTools[NumericDsolveMiddle]` по интервалу $\tau \in [t - 2\pi, t + 2\pi]$:

```
> CM:=(t)->DifEqTools[NumericDsolveMiddle](EqF(0),ICF,[method=dverk78,maxfun=500000,
  abserr = 1.*10^(-6), relerr = 1.*10^(-6)],tau,Phi(tau),4*Pi,t,100,S);
> GM_middle:=plot(CM(t),t=0..100,labels=[tau,Phi_0(tau)],label font=[TIMES,ROMAN,14],
  color=black);

> CMZ:=(t)->DifEqTools[NumericDsolveMiddle](EqF(0),ICF,[method=dverk78,maxfun=500000,
  abserr = 1.*10^(-6), relerr = 1.*10^(-6)],tau,D(Phi)(tau),4*Pi,t,100,S);
> GMZ_middle:=plot(CMZ(t),t=0..100,labels=[tau,Z_0(tau)],label font=[TIMES,ROMAN,14],
  color=black);
```

На рисунках 43 и 44 представлены графики зависимости средних от космологического времени τ .

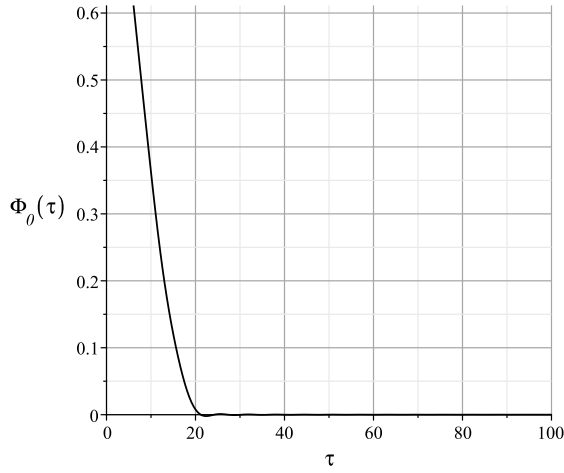


Рис. 43. Среднее значение потенциала скалярного поля, $\Phi(\tau)$, метод `dverk78`, время построения графика $\Delta t = 66.765$ сек.

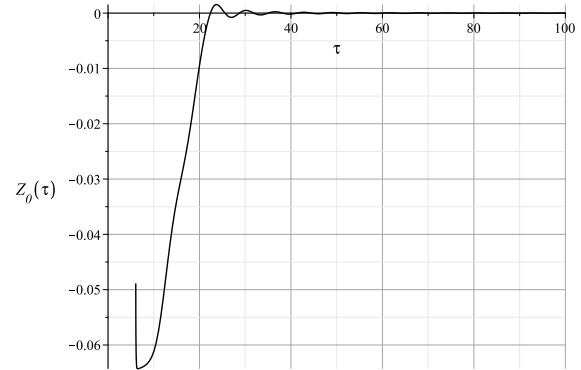


Рис. 44. Среднее значение производной потенциала скалярного поля, $Z = \dot{\Phi}(\tau)$, метод `dverk78`, время построения графика $\Delta t = 88,828$ сек.

Выделим теперь быстрые осцилляции динамических переменных $\delta\Phi(t)$ и $\delta Z(t)$ с помощью программных процедур усреднения `DifEqTools[NumericDsolveOsc]` по интервалу $\tau \in [t - 2\pi, t + 2\pi]$:

```
> CMF_Osc:=(t)->DifEqTools[NumericDsolveOsc](EqF(0),ICF,[method=dverk78,maxfun=500000,
abserr = 1.*10^(-6), relerr = 1.*10^(-6)],tau,Phi(tau)),4*Pi,t,100,S);

> CMZ_Osc:=(t)->DifEqTools[NumericDsolveOsc](EqF(0),ICF,[method=dverk78,maxfun=500000,
abserr = 1.*10^(-6), relerr = 1.*10^(-6)],tau,D(Phi)(tau)),4*Pi,t,100,S);

> GCMZ_Osc:=plot(CMZ_Osc(t),t=0..100,labels=[tau,delta Z],labelfont=[TIMES,ROMAN,14],
color=black);
```

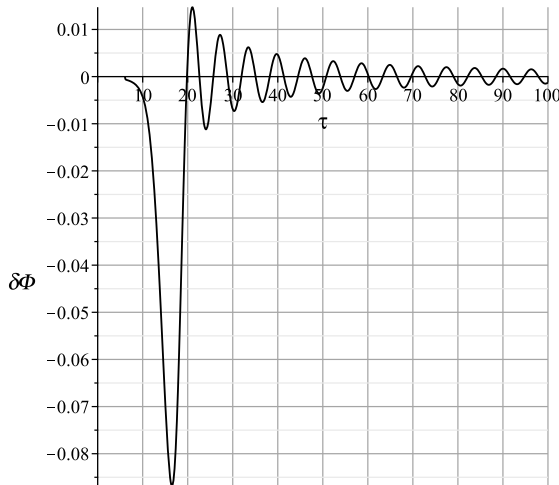


Рис. 45. Осцилляции потенциала скалярного поля, $\delta\Phi(\tau)$, метод `dverk78`, время построения графика $\Delta t = 115.922$ сек.

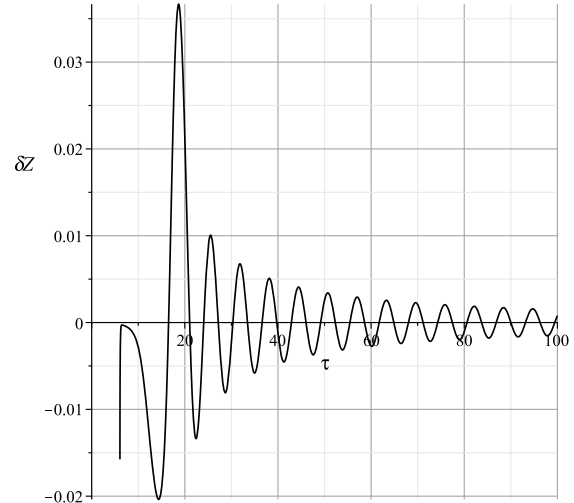


Рис. 46. Осцилляции производной потенциала скалярного поля, $\delta Z(\tau)$, метод `dverk78`, время построения графика $\Delta t = 185.656$ сек.

Вычислим теперь среднюю плотность энергии скалярного поля и среднее отношения — *среднее значение показателя баротропы*:

$$\kappa = \left\langle \frac{p}{\varepsilon} \right\rangle \quad (34)$$

по формулам (29).

```
> CME_s:=(t)->DifEqTools[NumericDsolveMiddle](EqF(0),ICF,[method=dverk78,maxfun=500000,
  abserr = 1.*10^(-6), relerr = 1.*10^(-6)],[tau,1/(8*Pi)*(D(Phi)(tau)^2+Phi(tau)^2)],
  4*Pi,t,100,S);
> T:=time():
  plot(log[10](CME_s(t)),t=0..1000,color=black,axes=BOXED,labels=[tau,epsilon__s(tau)]);
  time()-T;
> CMPE_s:=(t)->DifEqTools[NumericDsolveMiddle](EqF(0),ICF,[method=dverk78,maxfun=500000,
  abserr = 1.*10^(-6), relerr = 1.*10^(-6)],
  [tau,(D(Phi)(tau)^2-Phi(tau)^2)/(D(Phi)(tau)^2+Phi(tau)^2)],4*Pi,t,100,S);
> T:=time():
  plot(CMPE_s(t),t=0..1000,color=black,axes=BOXED,labels=[tau,p__s/epsilon__s]);
  time()-T;
```

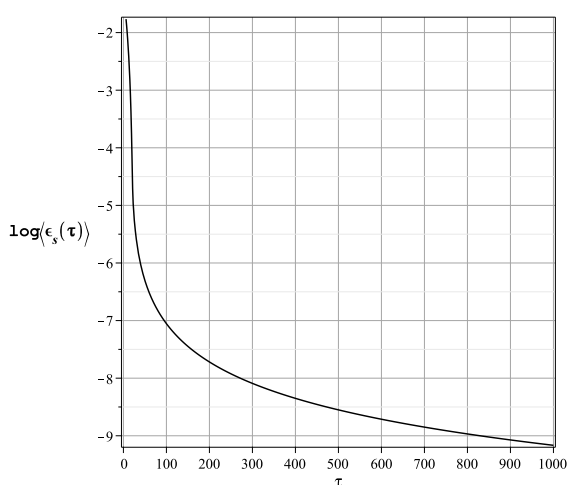


Рис. 47. Среднее значение плотности энергии скалярного поля.

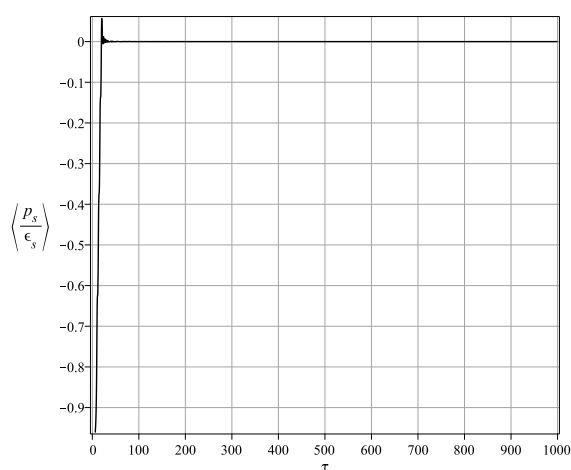


Рис. 48. Среднее значение показателя баротропы классического скалярного поля.

Как видно из графика на рисунке 48, показатель баротропы очень быстро выходит на нулевое значение. Это говорит, с одной стороны, о том, что с макроскопической точки зрения уравнение состояния скалярного поля становится нерелятивистским, а с другой стороны, — что средние значения квадрата потенциала и его производной со временем сближаются [10], [11]:

$$\langle \Phi^2(\tau) \rangle = \langle Z^2(\tau) \rangle, \quad \tau \rightarrow +\infty. \quad (35)$$

7. Заключение

Проведенные исследования показали важность развития численно – аналитических методов математического и компьютерного моделирования нелинейных динамических систем, в том числе, высокой размерности. В частности, применение этих методов позволило установить ряд новых свойств космологических моделей, основанных на скалярных полях, и эффективно их исследовать.

ЛИТЕРАТУРА

1. Игнатъев Ю.Г., Самигуллина А.Р. Численно – аналитические методы математического моделирования нелинейных динамических систем в СКМ Maple // Пространство, время и фундаментальные взаимодействия. 2016. Вып. 4(17). С. 78–105.

2. Игнатьев Ю.Г., Самигуллина А.Р. Численно – аналитические методы математического моделирования нелинейных динамических систем В СКМ Maple. II. Автоматизация математического анализа нелинейных математических моделей // Пространство, время и фундаментальные взаимодействия. 2017. Вып. 2(19). С. 74–83.
3. Игнатьев Ю.Г., Самигуллина А.Р. Программный комплекс численно – аналитического моделирования нелинейных динамических систем в СКМ Maple // Пространство, время и фундаментальные взаимодействия. 2016. Вып. 4(17). С. 147–148.
4. Игнатьев Ю.Г., Самигуллина А.Р. Программный комплекс численно-аналитического моделирования нелинейных динамических систем в СКМ Maple // Пакет программ.
URL: <http://www.stfi.ru/ru/software.html>.
5. Игнатьев Ю.Г. Математическое моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию. Казань: Казанский университет, 2014. 298 с.
6. Игнатьев Ю.Г., Агафонов А.А. Математические основы физики: Математические модели теоретической физики с примерами решения задач в СКМ Maple. Казань: Казанский университет, 2016. 264 с.
URL: http://kpfu.ru/staff_files/F1979419611/MMTF.pdf
7. Игнатьев Ю.Г., Агафонов А.А. Качественный и численный анализ космологической модели, основанной на фантомном скалярном поле с самодействием. II. Сравнительный анализ моделей с классическим и фантомным полями // Пространство, время и фундаментальные взаимодействия. 2017. Вып. 1(18). С. 46–65.
8. Журавлев В.М. Качественный анализ космологических моделей со скалярным полем // Пространство, время и фундаментальные взаимодействия. 2016. Вып. 4(17). С. 39–51.
9. Боголюбов Н.Н., Митропольский Ю.А. Собрание научных трудов. В 12 томах. Том 3. Математика и нелинейная механика. Асимптотические методы. М: Наука, 2005. 494 с.
10. Игнатьев Ю.Г., Самигуллина А.Р. Усреднение уравнений стандартной космологической модели по быстрым осцилляциям // Изв. вузов. Физика. 2017. Т. 60. № 7. С. 77–83.
11. Ignat'ev Yu.G., Ignatyev D.Yu., Samigullina A.R. A macroscopic view to the standard cosmological model // arXiv:1705.05000 [gr-qc].

Поступила в редакцию 16.07.2017

Игнатьев Юрий Геннадьевич, д. ф.-м. н., профессор, кафедра высшей математики и математического моделирования, Институт математики и механики им. Н.И. Лобачевского, Казанский федеральный университет, 420008, Россия, г. Казань, ул. Кремлевская, 35.

E-mail: ignat'ev_yu@rambler.ru

Самигуллина Алсу Ринатовна, ассистент, кафедра высшей математики и математического моделирования, Институт математики и механики им. Н.И. Лобачевского, Казанский федеральный университет, 420008, Россия, г. Казань, ул. Кремлевская, 35.

E-mail: alsu_sam@rambler.ru

Просьба ссылаться на эту статью следующим образом:

Игнатьев Ю.Г., Самигуллина А.Р. Численно-аналитические методы математического моделирования нелинейных динамических систем в СКМ Maple. III. Визуализация нелинейных динамических систем и приложения к механике и космологии // Пространство, время и фундаментальные взаимодействия. 2017. № 3. С. 28–56.

Yu. G. Ignat'ev, A. R. Samigullina

Numerically-analytical methods for mathematical modeling of nonlinear dynamic systems in CAS Maple. III. Visualization of nonlinear dynamics systems and applications to mechanics and cosmology

Keywords: nonlinear dynamical systems, mathematical modeling, qualitative theory of differential equations, spline approximation, numerical integration methods, computational visualization, nonlinear mechanics, cosmology, applied mathematical packages.

PACS: 02.60.Cb, 07.05.Tp, 04.25.-g

The tools for automatic visualization of numerical solutions of systems of nonlinear ordinary differential equations and averaging of numerical solutions on the basis of the author's application package of DifEqTools programs are described, and examples of application of these tools to the study of nonlinear problems in mechanics and cosmology are given.

REFERENCES

1. Ignat'ev Yu.G., Samigullina A.R. Numerically - analytical methods for mathematical modeling of nonlinear dynamic systems in CAS Maple, *Space, Time and Fundamental Interactions*, 2016, no. 4(17), pp. 78–105.
2. Ignat'ev Yu.G., Samigullina A.R. Numerically - analytical methods for mathematical modeling of nonlinear dynamic systems in CAS Maple. II. Automation of the mathematical analysis of nonlinear mathematical models, *Space, Time and Fundamental Interactions*, 2016, no. 2(19), pp. 74–83.
3. Ignat'ev Yu.G., Samigullina A.R. Program complex for numerical and analytical mathematical modeling of nonlinear dynamic systems in CAS Maple, *Space, Time and Fundamental Interactions*, 2016, no. 4(17), pp. 147–148.
4. Ignat'ev Yu.G., Samigullina A.R. The software package numerical-analytical modeling of nonlinear dynamic systems in the SCM Maple, *Space, Time and Fundamental Interactions*.
<http://www.stfi.ru/ru/software.html>
5. Ignatyev Yu.G. *Matematicheskoe modelirovanie fundamental'nykh objektov i yavlenij v sisteme komp'yuternoj matematiki Maple. Lekcii dlya shkoly po matematicheskomu modelirovaniyu* (Mathematical modeling of fundamental objects and phenomena in system of computer mathematics Maple. Lectures for school on mathematical modeling), Kazan: Kazan University, 2014, 298 p.
6. Ignatyev Yu.G., Agathonov A.A. *Matematicheskie osnovy fiziki: matematicheskie modeli teoreticheskoy fiziki s primerami resheniya zadach v SKM Maple* (Mathematical foundations of physics: Mathematical models of theoretical physics with examples of solving problems in SCM Maple), Kazan: Kazan University, 2016, 264 p.
http://kpfu.ru/staff_files/F1979419611/MMTF.pdf
7. Ignat'ev Yu.G., Agathonov A.A. Qualitative and numerical analysis of the cosmological model based on the phantom scalar field with self-interaction. II. Comparative analysis of models of classical and phantom fields, *Space, Time and Fundamental Interactions*, 2017, no. 1(18), pp. 46–65.
8. Zhuravlev V.M. Qualitative analysis of cosmological models with scalar fields, *Space, Time and Fundamental Interactions*, 2016, no. 4(17), pp. 39–51.
9. Bogolyubov N.N., Mitropolsky Yu.A. *Matematika i nelinejnaya mekhanika. Asimptoticheskie metody* (Mathematics and nonlinear mechanics. Asymptotic methods), Moscow: Science, 2005, 494 p.
10. Ignat'ev Yu.G., Samigullina A.R. Averaging of the equations of the standard cosmological model over fast oscillations, *Izvestiya Vysshikh Uchebnykh Zavedenii. Fizika*, 2017, vol. 60, no. 7, pp. 77–83.
11. Ignat'ev Yu.G., Ignatyev D.Yu., Samigullina A.R. A macroscopic view to the standard cosmological model, *arXiv:1705.05000 [gr-qc]*.

Received 16.07.2017

Ignat'ev Yurii Gennadievich, Doctor of Physics and Mathematics, Professor, Lobachevsky Institute of Mathematics and Mechanics, Kazan Federal University, ul. Kremlyovskaya, 35, Kazan, 420008, Russia.

E-mail: ignat'ev_yu@rambler.ru

Samigullina Alsu Rinatovna, assistant lecturer, Lobachevsky Institute of Mathematics and Mechanics, Kazan Federal University, ul. Kremlyovskaya, 35, Kazan, 420008, Russia.

E-mail: alsu_samgrambler.ru

Please cite this article in English as:

Ignat'ev Yu. G., Samigullina A. R. Numerically-analytical methods for mathematical modeling of nonlinear dynamic systems in CAS Maple. III. Visualization of nonlinear dynamics systems and applications to mechanics and cosmology, *Space, Time and Fund. Interact.*, 2017, no. 3, pp. 28–56.