

AUTOMATED VISUALIZATION OF NON-LINEAR DYNAMIC SYSTEMS AND ITS APPLICATIONS
TO MECHANICS AND COSMOLOGY

Yu.G. Ignatev, A.R. Samigullina

Author's program procedures for automated research and visualization of nonlinear dynamical systems and their applications to the problems of mechanics and cosmology are described. In particular, procedures of the automated averaging of rapidly oscillating numerical solutions are described.

Keywords: nonlinear dynamical systems, mathematical modeling, qualitative theory of differential equations.

УДК 514.822; 27.35

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

А.Р. Самигуллина¹

¹ *alsu_sam@mail.ru*; Казанский (Приволжский) федеральный университет

Описаны инструменты автоматизированного решения и визуализации численных решений систем нелинейных обыкновенных дифференциальных уравнений произвольного порядка, разрешенных относительно старших производных.

Ключевые слова: нелинейные динамические системы, математическое моделирование, качественная теория дифференциальных уравнений.

Программные процедуры автоматизированной визуализации нелинейных динамических систем в пакете DifEqTools

1. Сервисные программные процедуры автоматизированного построения графиков функций численных решений

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

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

где $y^{(n)} = d^n y / dt^n$ – обозначение n -той производной функции $y(t)$ по независимой переменной t , временной переменной¹, а F_i – непрерывно-дифференцируемые функции своих переменных. Будем в дальнейшем полагать выполненными начальными условиями 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 $y_i(t)$.

¹ Хотя это может быть и не обязательно физическое время.

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

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

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

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 ([1]) 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)];
```

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

```
> 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, 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.

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

```
> 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);
```

Продемонстрируем 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];
```

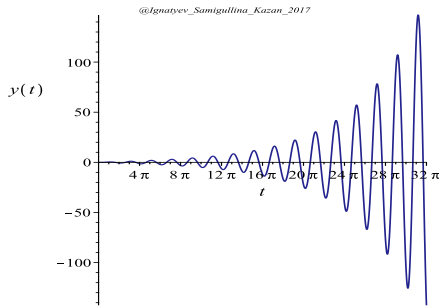


Рис. 1. Автоматизированный вывод численного решения задачи Коши (5) – (6): выводится нулевая производная второй неизвестной функции $y(t)$ на отрезке $[0, 32\pi]$.

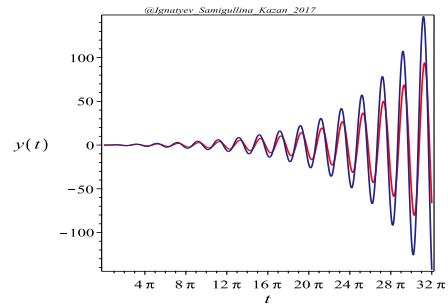


Рис. 2. Совмещение графиков двух численных решений задачи Коши с начальными условиями ICS и ICS_1 с помощью команды `plots[display]`.

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

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

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);
```

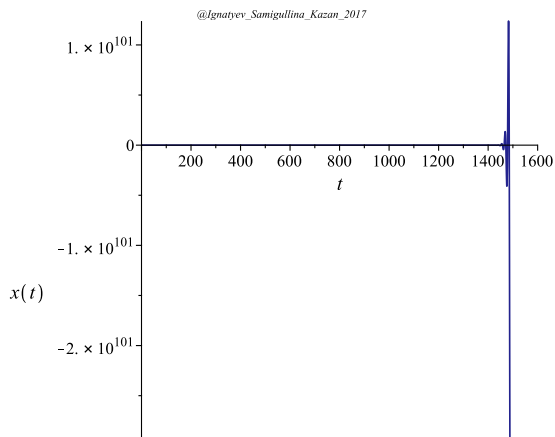


Рис. 3. Автоматизированный вывод численного решения задачи Коши (7) – (8) при параметре $L=1600$.

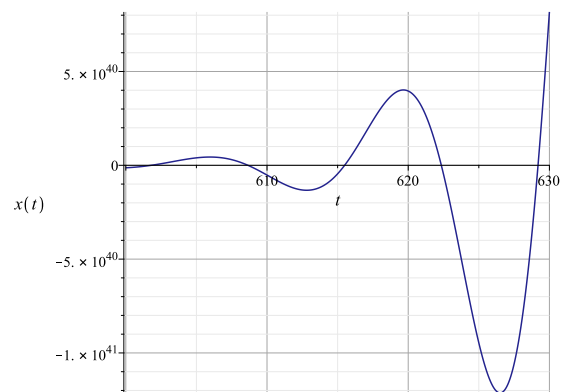


Рис. 4. Автоматизированный вывод численного решения задачи Коши (7) – (8) при параметре $L=[600, 630]$.

2. 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

ке L в виде линии цвета $Color$ реализуется программной процедурой $GraphPhase2d(Eqs, ICS, Method, [[i1, j1], [i2, j2]], L, Color)$ ([2]). При этом автоматически помечаются зеленой и красной точками начало и конец фазовой траектории.

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

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

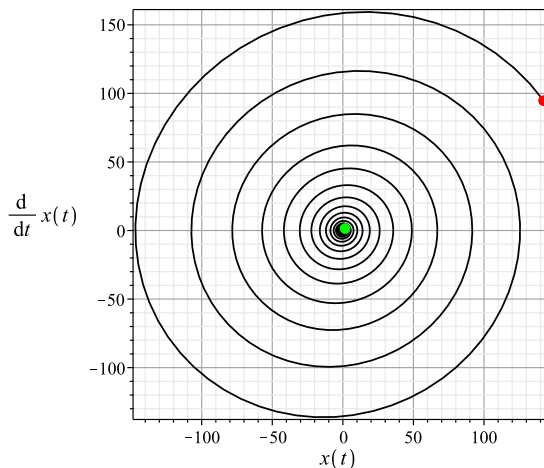


Рис. 5. Автоматизированный вывод фазовой траектории численного решения задачи Коши (5) – (6) $x(t), dx(t)/dt$ на отрезке $[0, 32\pi]$.

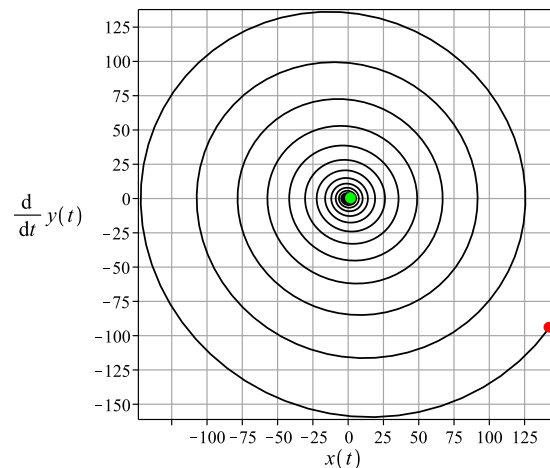


Рис. 6. Автоматизированный вывод фазовой траектории численного решения задачи Коши (5) – (6) $x(t), dy(t)/dt$ на отрезке $[0, 32\pi]$.

Автоматизированный вывод трехмерной фазовой траектории численного решения методом $Method$ задачи Коши $Eqs - ICS$ на отрезке L в виде линии цвета $Color$ реализуется программной процедурой

$GraphPhase3d(Eqs, ICS, Method, [[i1, j1], [i2, j2], [i3, j3]], L, Color)$, 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];
```

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

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

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

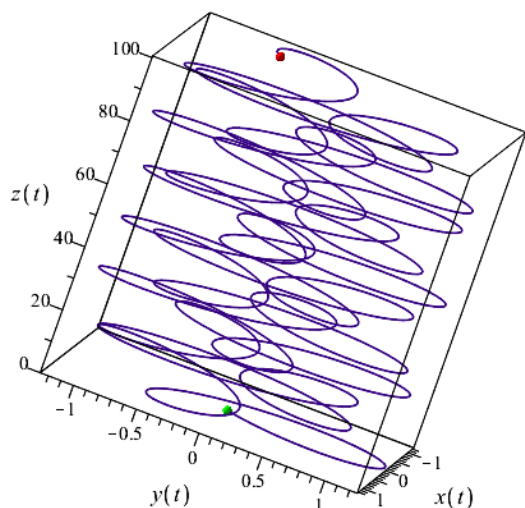


Рис. 7. Автоматизированный вывод фазовой траектории численного решения задачи Коши (9) – (10) методом rkf45 в форме фазовой траектории: $[x(t), y(t), z(t)]$ на отрезке $[0, 32\pi]$.

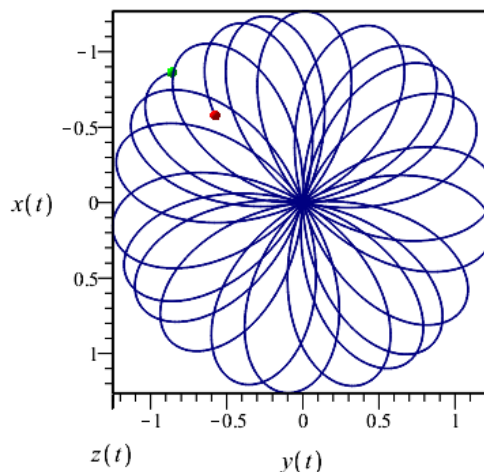


Рис. 8. Зеленая точка – начало траектории, красная – ее конец.

Выше мы отмечали, что описанные программные процедуры с одинаковым успехом получают и реализуют численные решения, как линейных, так и нелинейных систем ОДУ. Продemonстрируем 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];
```

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

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

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`

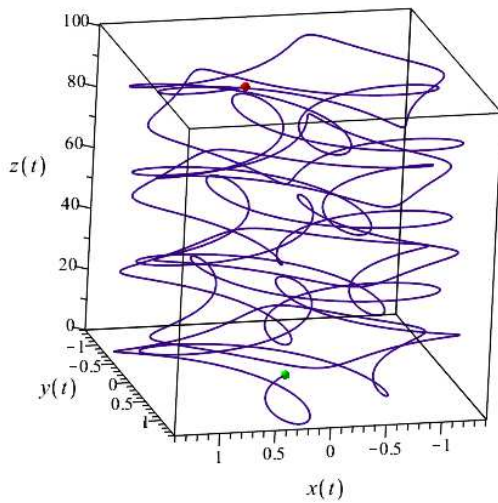


Рис. 9. Автоматизированный вывод численного решения задачи Коши (11), (10) методом rkf45 в форме фазовой траектории: $[x(t), y(t), z(t)]$ на отрезке $[0, 32\pi]$.

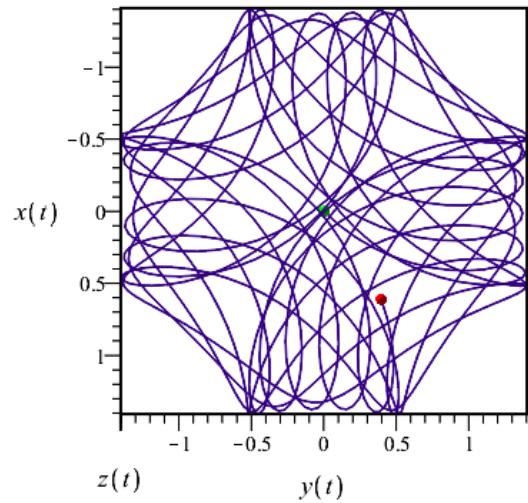


Рис. 10. Проекция фазовой траектории Рис. 9 на плоскость $\{XOY\}$ (углы Эйлера: $\text{orientation} = [\theta = 0, \psi = 0, \varphi = 0]$). Зеленая точка – начало траектории, красная – ее конец.

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 равный 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];
```

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

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

```
> 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);
```

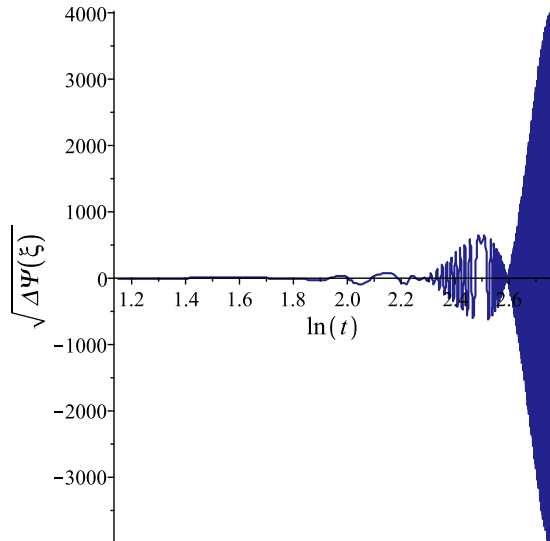


Рис. 11. Автоматизированный вывод численного решения задачи Коши (12), (6) методом rkf45 $[\ln t, \sqrt{x(t)^2} \sin(y(t))]$ на отрезке $[\pi, 5\pi]$. На оси ординат выведено имя пользовательской функции $\sqrt{\Delta\Psi(\xi)}$.

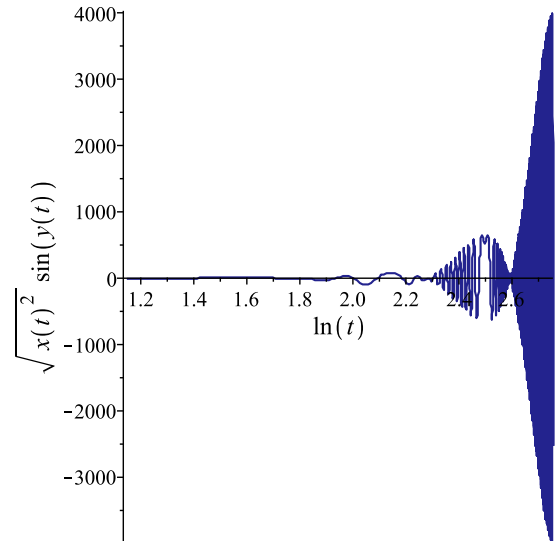


Рис. 12. Автоматизированный вывод численного решения задачи Коши (12), (6) методом rkf45 $[\ln t, \sqrt{x(t)^2} \sin(y(t))]$ на отрезке $[\pi, 5\pi]$. На оси ординат выведено истинное имя функции $\sqrt{x(t)^2} \sin(y(t))$.

Аналогично устроены и команды `DifEqTools[GraphPhase2dFun]` и `DifEqTools[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.

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

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

```
> 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 системах координат:

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

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);
```

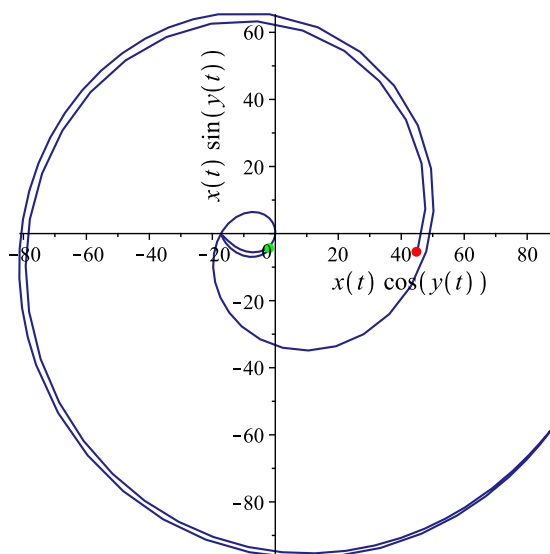


Рис. 13. Автоматизированный вывод численного решения задачи Коши (5) – (6) в форме двумерной фазовой траектории в полярной системе координат на отрезке $[\pi, 3\pi]$.

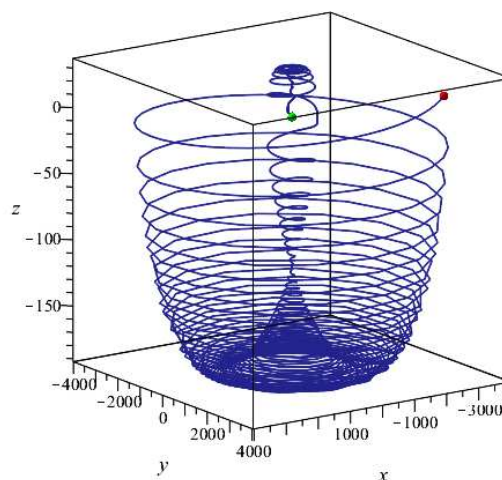


Рис. 14. Автоматизированный вывод численного решения задачи Коши (5) – (6) в форме трехмерной фазовой траектории в цилиндрической системе координат на отрезке $[0, 5\pi]$.

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

Пакет DifEqTools ([3]-[4]) 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 ([5]) 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:

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

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

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

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

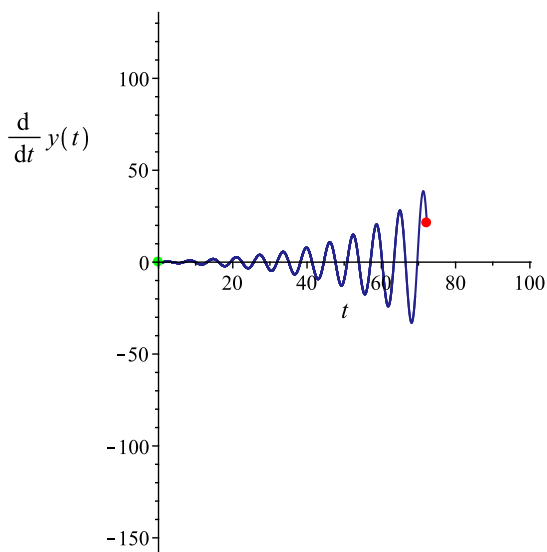


Рис. 15. 24-й кадр 32-х кадровой анимации численного решения задачи Коши (5) – (6) в форме конфигурационной траектории $\dot{y}(t)$ на отрезке $[0, 32\pi]$.

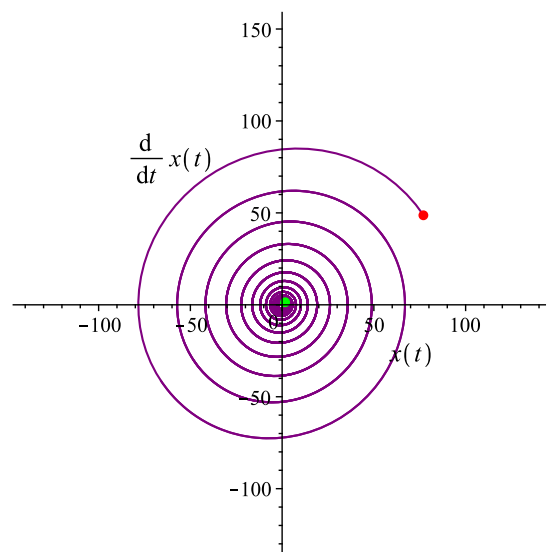


Рис. 16. 28-й кадр 32-х кадровой анимации численного решения задачи Коши (5) – (6) в форме фазовой траектории в плоскости $\{x(t), \dot{x}(t)\}$ на отрезке $[0, 32\pi]$.

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.

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

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

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

Таким образом, при сокращении числа точек графика в 3,3 раза скорость создания анимационной модели увеличивается в 3 раза. При этом не происходит за-

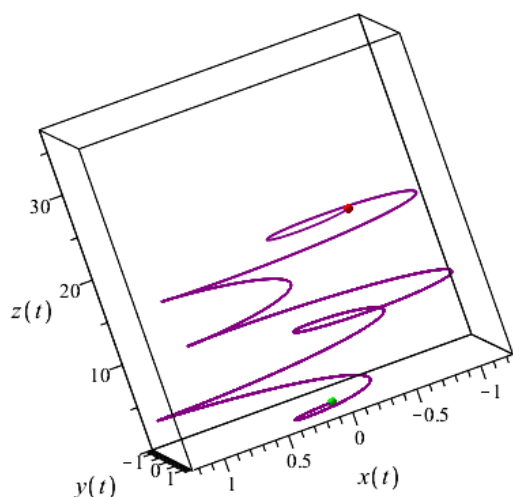


Рис. 17. 20-й кадр 32-х кадровой 3d-анимации численного решения задачи Коши (9) – (10) на отрезке $[0, 12\pi]$.

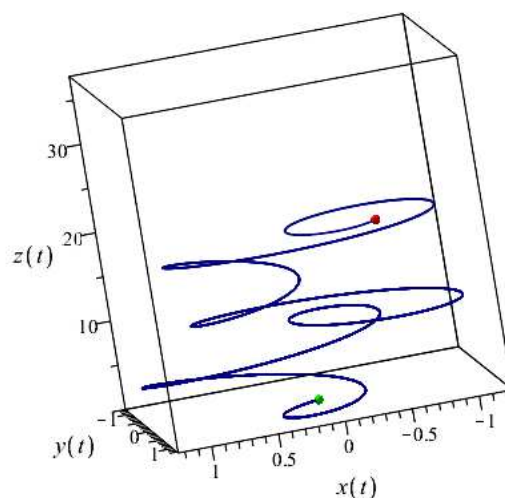


Рис. 18. 20-й кадр 32-х кадровой 3d-анимации численного решения задачи Коши (9) – (10) на отрезке $[0, 12\pi]$. Использован необязательный параметр `numpoints=300`.

метного 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.

Литература

1. Игнатъев Ю.Г. Численно - аналитические методы математического моделирования нелинейных динамических систем в СКМ Maple / Ю.Г. Игнатъев, А.Р. Самигуллина // Пространство, время и фундаментальные взаимодействия. – 2016. – Вып. 4(17). – С. 78–105.
2. Игнатъев Ю.Г. Численно - аналитические методы математического моделирования нелинейных динамических систем В СКМ Maple. II. Автоматизация математического анализа нелинейных математических моделей / Ю.Г. Игнатъев, А.Р. Самигуллина // Пространство, время и фундаментальные взаимодействия. – 2017. – Вып. 2(19). – С. 74–83.
3. Игнатъев Ю.Г. Программный комплекс численно - аналитического моделирования нелинейных динамических систем в СКМ Maple / Ю.Г. Игнатъев, А.Р. Самигуллина // Пространство, время и фундаментальные взаимодействия. – 2016. – Вып. 4(17). – С. 147–148.
4. Игнатъев Ю.Г. Программный комплекс численно-аналитического моделирования нелинейных динамических систем в СКМ Maple / Ю.Г. Игнатъев, А.Р. Самигуллина // Пакет программ. – Режим доступа: <http://www.stfi.ru/ru/software.html>.
5. Игнатъев Ю.Г. Математическое моделирование фундаментальных объектов и явлений в системе компьютерной математики Maple. Лекции для школы по математическому моделированию / Ю.Г. Игнатъев, А.Р. Самигуллина. – Казань: Казанский университет, 2014. – 298 с.

NUMERICALLY-ANALYTICAL METHODS FOR MATHEMATICAL MODELING
OF NONLINEAR DYNAMIC SYSTEMS IN CAS MAPLE

A.R. Samigullina

The tools of the automated solution and visualization of numerical solutions of systems of nonlinear ordinary differential equations of arbitrary order solved with respect to the highest derivatives are described.

Keywords: nonlinear dynamical systems, mathematical modeling, qualitative theory of differential equations.

УДК 5530.12+531.51

**КАЧЕСТВЕННЫЙ АНАЛИЗ КОСМОЛОГИЧЕСКИХ МОДЕЛЕЙ С
МИНИМАЛЬНЫМ КЛАССИЧЕСКИМ И ФАНТОМНЫМ СКАЛЯРНЫМ
ВЗАИМОДЕЙСТВИЕМ**Ю.Г. Игнатьев¹, А.А. Агафонов²

¹ ignatev-yurii@mail.ru; Казанский (Приволжский) федеральный университет

² a.a.agathonov@gmail.com; Казанский (Приволжский) федеральный университет

На основе качественного анализа космологических моделей с классическими и фантомными скалярными полями с самодействием произведена классификация особых точек динамических систем.

Ключевые слова: космологические модели, классическое и фантомное скалярное поле, качественный анализ.

1. Введение

Стандартные космологические модели (СКМ), основанные на классическом скалярном поле, были исследованы методами качественного анализа динамических систем в работах [1], [2], [3], [4], [5], [6]. В работах одного из Авторов неминимальная теория скалярного взаимодействия на основе понятия фундаментального скалярного заряда последовательно развивалась, как для классических, так и фантомных скалярных полей [7], [8], [9], [10]. Указанные исследования показывают необходимость исследования фантомных скалярных полей в качестве возможной основы космологической модели ранней Вселенной.

В настоящей работе в соответствии с общепринятой терминологией мы будем называть фантомными скалярные поля с отрицательным кинетическим членом в тензоре энергии-импульса независимо от знака потенциального члена. При этом фантомным скалярным полям с притяжением одноименно заряженных частиц соответствует отрицательный потенциальный член в тензоре энергии - импульса, а фантомным полям с отталкиванием - положительный потенциальный член. В первом случае знаки кинетического и потенциального члена совпадают, во втором - противоположны, что эквивалентно изменению знака массивного члена в уравнении Клейна - Гордона. Соответствующие решения для уединенного скалярного за-