

Оглавление

Постановка задачи

Методы решения

Проверка корректности методов

Применение методов к поставленной задаче, анализ результатов

Постановка задачи

Дано:

$$\begin{cases} x' = -a\left(\frac{x^3}{3} - x\right) + ay, \\ y' = -x - by + c, \\ x(0) = 2, \quad y(0) = 0 \end{cases}$$

Здесь $1 \leq a \leq 10^3$, $0 < c < 1$.

Задача: Провести исследование поведения решений в зависимости от значений «большого» параметра a .

Методы решения

Для поиска первообразных было выбрано пять численных метода: *явный метод Эйлера*, *явный метод Рунге-Кутты 4го и 5го порядка*, *неявный метод Эйлера 1го порядка* и *неявный метод Рунге-Кутты 2го порядка (приближение трапецией)*.

Явный метод Эйлера

Имеем отрезок времени $[T_0, T_{\text{fin}}]$. Разобьем его точками t_0, t_1, \dots, t_n , где $t_1 = T_0$, $t_n = T_{\text{fin}}$ и $\tau = t_i - t_{i-1}$. Для каждой i -ой точки найдем (x_i, y_i) ($0 \leq i \leq n$). Имеем $\frac{x_i - x_{i-1}}{\tau} = x(x_{i-1}, y_{i-1})$ и $\frac{y_i - y_{i-1}}{\tau} = y(x_{i-1}, y_{i-1})$ ($1 \leq i \leq n$). Получим уравнения:

$$x_i = \begin{cases} x_0 & , \text{ при } i = 0 \\ x(x_{i-1}, y_{i-1})\tau + x_{i-1} & , \text{ иначе} \end{cases}$$

и

$$y_i = \begin{cases} y_0 & , \text{ при } i = 0 \\ y(x_{i-1}, y_{i-1})\tau + y_{i-1} & , \text{ иначе} \end{cases}$$

Применение его к данной задаче:

$$x_i = \begin{cases} 2 & , \text{ при } i = 0 \\ \left(-a\left(\frac{x_{i-1}^3}{3} - x_{i-1}\right) + ay_{i-1}\right)\tau + x_{i-1} & , \text{ иначе} \end{cases}$$

и

$$y_i = \begin{cases} 0 & , \text{ при } i = 0 \\ (-x_{i-1} - by_{i-1} + c)\tau + y_{i-1} & , \text{ иначе} \end{cases}$$

По этим значениям и будем строить график.

Неявный метод Эйлера

Неявный метод Эйлера, как и явный, имеет первый порядок, но отличается от последнего тем, что для его применения требуется решить систему уравнений:

$$\begin{cases} x_i = x(x_i, y_i)\tau + x_{i-1} \\ y_i = y(x_i, y_i)\tau + x_{i-1} \end{cases}$$

Применений к нашей задаче:

$$\begin{cases} x_i = (-a(\frac{x_i^3}{3} - x_i) + ay_i)\tau + x_i \\ y_i = (-x_i - by_i + c)\tau + y_i \end{cases}$$

Будем численно искать решение этой системы с помощью *метода Ньютона*.

С помощью него будем искать x_i . Рассмотрим функцию

$f(x_i) = -x_i + (-a(\frac{x_i^3}{3} - x_i) + ay_i)\tau + x_i = 0$. Суть метода ньютона найти такое \tilde{x}_n , такую что $f(\tilde{x}_n) \approx 0$, где n - число итераций алгоритма, а \tilde{x}_n - точка приближения. Зададим начальную точку приближения $\tilde{x}_0 = x_{i-1}$.

$$\tilde{x}_i = \tilde{x}_{i-1} - \frac{f(\tilde{x}_{i-1})}{f'(\tilde{x}_{i-1})} (1 \leq i \leq n)$$

Таким образом находим n -ую точку приближения и считаем, что она равна x_i . Чем больше итераций будет сделано, тем ближе эта точка будет к настоящему решению, но делать их слишком много тоже не имеет смысла.

После того, как мы найдем таким образом x_i подставляем ее в уравнение для y_i . Так мы получаем точки (x_i, y_i) , по ним и строим график.

Метод Рунге-Кутты 4-го порядка

Метод Рунге-Кутты 4-го порядка - явный одношаговый метод. Точность возрастает за счет дополнительных вычислений на отрезке $[t_n, t_n + T]$. Если положить $\frac{dy}{dt} = f(t, y)$, где t - независимая переменная, то алгоритм выглядит так:

$$k_1 = f(t_n, y_n),$$

$$k_2 = f(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_1),$$

$$k_3 = f(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_2),$$

$$k_4 = f(t_n + \tau, y_n + \tau k_3)$$

тогда следующее значение $y(t)$ считается по формуле:

$$y_{n+1} = y_n + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

В поставленной задаче фактически три переменных, поэтому, если взять $\frac{dx}{dt} = f_x(t, x, y)$ и $\frac{dy}{dt} = f_y(t, x, y)$, метод примет следующий вид:

$$k_{y1} = f_y(t_n, y_n, x_n),$$

$$k_{x1} = f_x(t_n, y_n, x_n),$$

$$k_{y2} = f_y(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y1}, x_n + \frac{\tau}{2}k_{x1}),$$

$$k_{x2} = f_x(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y1}, x_n + \frac{\tau}{2}k_{x1}),$$

$$k_{y3} = f_y(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y2}, x_n + \frac{\tau}{2}k_{x2}),$$

$$k_{x3} = f_x(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y2}, x_n + \frac{\tau}{2}k_{x2}),$$

$$k_{y4} = f_y(t_n + T, y_n + \tau k_{y3}, x_n + \tau k_{x3}),$$

$$k_{x4} = f_x(t_n + T, y_n + \tau k_{y3}, x_n + \tau k_{x3}),$$

соответственно:

$$y_{n+1} = y_n + \frac{\tau}{6}(k_{y1} + 2k_{y2} + 2k_{y3} + k_{y4}),$$

$$x_{n+1} = x_n + \frac{\tau}{6}(k_{x1} + 2k_{x2} + 2k_{x3} + k_{x4})$$

Метод Рунге-Кутты-Фельберга 5-го порядка

Методы из семейства Рунге-Кутты могут быть представлены через таблицу Бутчера

c_1	a_{11}	a_{12}	\dots	a_{1s}
c_2	a_{21}	a_{22}	\dots	a_{2s}
\vdots	\vdots	\vdots	\ddots	\vdots
c_s	a_{s1}	a_{s2}	\dots	a_{ss}
	b_1	b_2	\dots	b_s

Явные методы имеют общий вид

$$\begin{aligned}
 k_1 &= f(t_n, y_n), \\
 k_2 &= f(t_n + c_2 h, y_n + h(a_{21} k_1)), \\
 k_3 &= f(t_n + c_3 h, y_n + h(a_{31} k_1 + a_{32} k_2)), \\
 &\vdots \\
 k_i &= f(t_n + c_i h, y_n + h \sum_{j=1}^{i-1} a_{ij} k_j), \\
 y_{n+1} &= y_n + h \sum_{i=1}^s b_i k_i
 \end{aligned}$$

Так как им соответствует ступенчатая таблица (вот почему суммирование до $i - 1$), то есть их применение не требует решения системы уравнений. Поэтому легко реализовать универсальную функцию, которая применяет метод по формулам, используя коэффициенты из таблицы.

Таблиц для метода одного порядка может быть несколько, но они должны удовлетворять некоторым условиям.

Так выглядит одна из таблиц Бутчера для явного метода Рунге-Кутты-Фельберга:

0						
$\frac{1}{4}$	$\frac{1}{4}$					
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$				
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$			
1	$\frac{1}{4}$	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$	
	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$

Неявный метод 2-го порядка

Для неявных методов, конечно, тоже существуют таблицы Бутчера. Но так как они не имеют нижнетреугольного вида, общая формула вычисления немного меняется:

$$k_i = f(t_n + c_i h, y_n + \tau \sum_{j=1}^s a_{ij} k_j) \quad (\text{суммирование до } s)$$

В проекте использовался *метод Кранка-Николсона* — правило трапеции, только в неявной форме. Таблица Бутчера для него следующая

0	0	0
1	$\frac{1}{2}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$

$$\begin{aligned} k_{y1} &= f_y(t_n, y_n, x_n), \\ k_{x1} &= f_x(t_n, y_n, x_n), \end{aligned}$$

Теперь требуется решить систему уравнений для двух неизвестных: k_{x2}, k_{y2}

$$\begin{cases} k_{y2} = f_y(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y1} + \frac{\tau}{2}k_{y2}, x_n + \frac{\tau}{2}k_{x1} + \frac{\tau}{2}k_{x2}), \\ k_{x2} = f_x(t_n + \frac{\tau}{2}, y_n + \frac{\tau}{2}k_{y1} + \frac{\tau}{2}k_{y2}, x_n + \frac{\tau}{2}k_{x1} + \frac{\tau}{2}k_{x2}) \end{cases}$$

Система уравнений в проекте решается с помощью модуля `SciPy` языка `Python`, конкретно, функцией `scipy.optimize.root(fun, x0, jac, method)`. Эта функция решает уравнение с помощью численных методов, принимая на вход

`fun` — функции, которые обращаются в ноль при подстановке корней

`x0` — предполагаемое решение, от которого отталкивается метод

`jac` — матрица Якоби для функций из `fun`

`method` — метод, которым будет решаться уравнение, был выбран `hybr`

После получения корней, т.е. k_{x2}, k_{y2} , остается подставить их в выражения

$$y_{n+1} = y_n + \tau \sum_{i=1}^s b_i k_i = y_n + \frac{\tau}{2}(k_{y1} + k_{y2}), \quad x_{n+1} = x_n + \tau \sum_{i=1}^s b_i k_i = x_n + \frac{\tau}{2}(k_{x1} + k_{x2})$$

Адаптивный шаг

В задачах такого типа зачастую шаг делают не фиксированным, а адаптивным, то есть меняющимся во время работы алгоритма. Делается это от того, что жесткие задачи содержат участки быстрого изменения решения, поэтому при взятии большого шага будет значительно теряться точность. Если же

брать постоянный маленький шаг алгоритм будет работать слишком долго. Мы также решили реализовать адаптивный шаг.

Мы имеем систему ОДУ, вида

$$\frac{f_i}{dt} = f_i(\mathbf{u}, t) \quad (1 \leq i \leq 2)$$

Возьмем новый аргумент - длину дуги интегральной кривой:

$$dl = \rho dt, \rho = \left(1 + \sum f_i^2\right)^{1/2},$$

тогда

$$\frac{f_i}{dl} = F_i(\mathbf{u}), F_i = f_i \rho^{-1} (1 \leq i \leq 2)$$

Будем выбирать шаг в зависимости от кривизны интегральной кривой, по формуле:

$$\tau = \frac{\tau_0}{(1 + L^{0.5/\nu}(\chi, \chi)^{0.25/\nu})^\nu}$$

Здесь L - полный промежуток интегрирования от 1, χ - кривизна, T_0 - шаг на почти прямом участке интегральной кривой. Численные расчеты показали, что для сверхжестких задач целесообразно выбирать $\nu = 2$, а для задач меньшей жесткости - $\nu = 1$. Мы взяли $\nu = 1$. Таким образом мы строим сетку, начиная с T_0 . Делается это одновременно с вычислением решения, поэтому она будет под него адаптирована. $\chi = T * prev(\mathbf{F} - \mathbf{F} * prev)$, где $T * prev$ - предыдущий шаг. Алгоритмы от применения адаптивного шага не меняются, меняется только сетка точек, по которым мы строим график.

Проверка корректности методов

Далее будет показано, как численные методы находят решение и какова зависимость погрешности вычислений от шага.

Для численных методов их ошибка зависит от шага интегрирования. Примем функцию $Y(t)$ за настоящее решение ДУ, а $y(t)$ за приближенное, тогда функция ошибки от шага

$$E(\tau) = Y(t) - y(t) \leq C\tau^n,$$

$$E(\tau) = O(\tau^n),$$

где n — порядок численного метода. Тогда прологарифмировав обе части неравенства получим

$$\ln(E(\tau)) = n * O(\ln(\tau))$$

То есть в координатах $\ln(E(\tau))$, $\ln(\tau)$ график будет прямой, **коэффициент наклона** которой и есть **порядок метода**.

1ая тестовая задача (не жесткая)

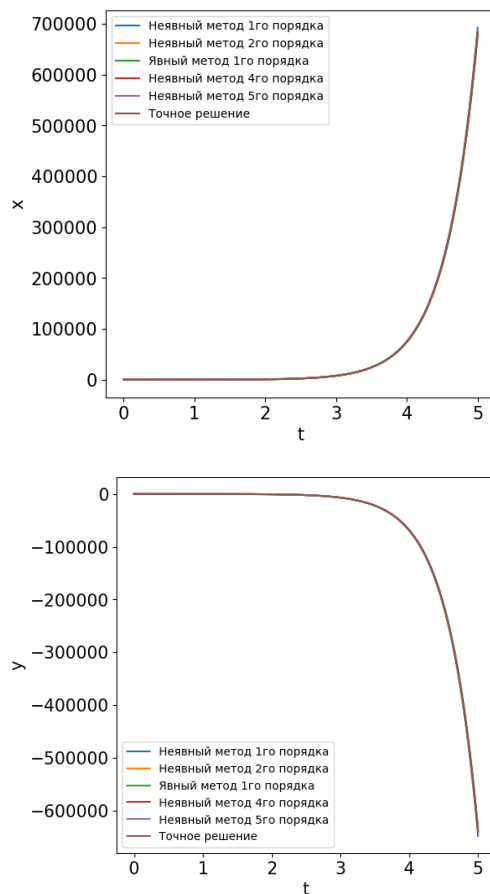
Для задачи

$$\begin{cases} x' = 5x + 3y, \\ y' = -3x - y \end{cases}$$

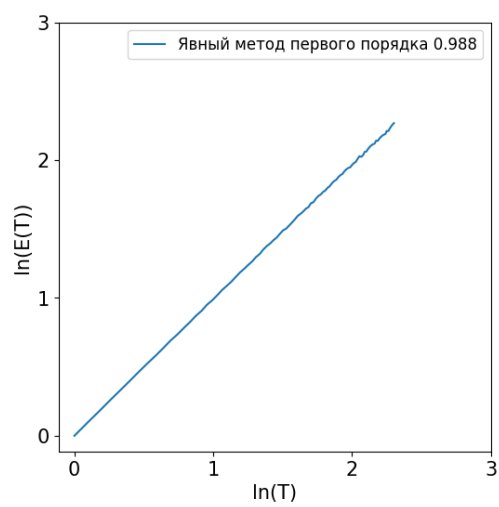
Точным решением является

$$\begin{cases} x = (1 + 6t) \exp(2t), \\ y = (1 - 6t) \exp(2t) \end{cases}$$

Методы приближают задачу следующим образом

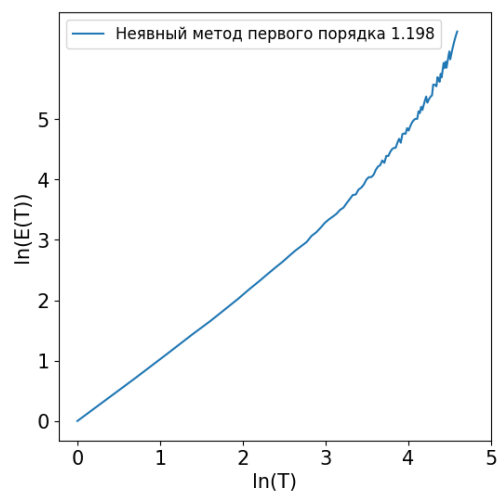


Явный метод Эйлера 1-го порядка

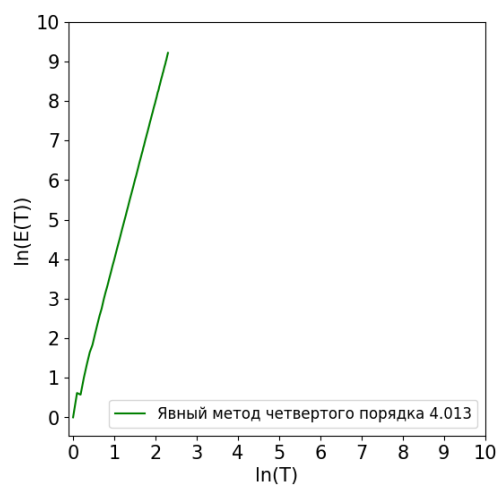


Видно, что порядок метода $\$ = 1\$$

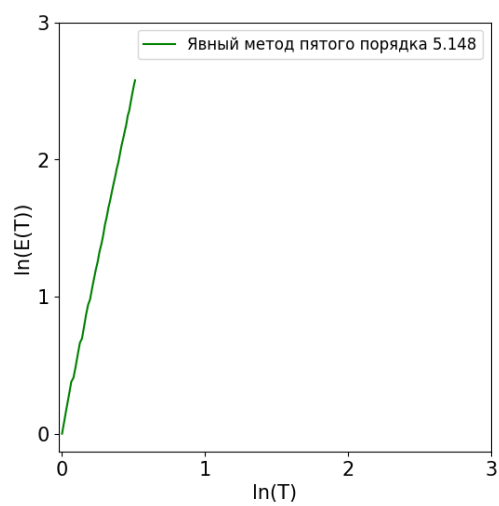
Неявный метод Эйлера 1-го порядка



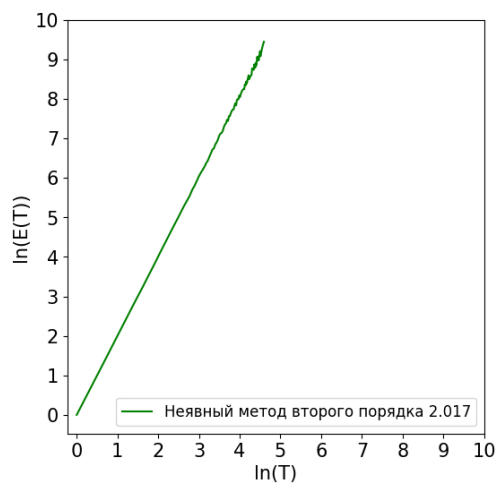
Метод Рунге-Кутты 4-го порядка



Метод Рунге-Кутты-Фельберга 5-го порядка



Метод Кранка-Николсона 2-го порядка



2ая тестовая задача (жесткая)

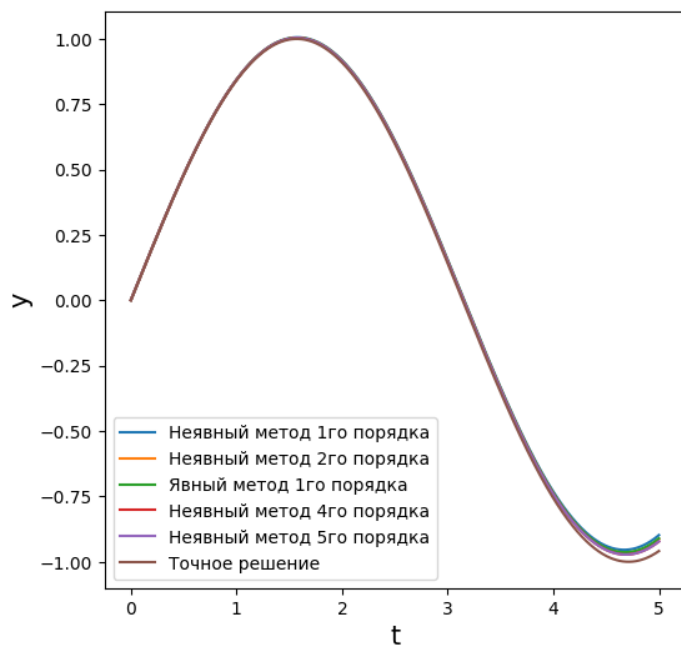
Наша вторая тестовая задача является жесткой. Общепринятого математического определения жестких ОДУ нет, однако у них есть важное свойство, явные методы являются крайне не устойчивыми при решении ими таких задач. Вот сама система:

$$\begin{aligned} x' &= \lambda [\cos^2(t) \sin(t) + 2 \cos(t) - (2 + xy)x] - y \\ y' &= x + y - \sin(t) \end{aligned}$$

Ее жесткость определяется фиксированным параметром λ . Эта задача имеет точное решение в замкнутой форме:

$$\begin{aligned} x &= \cos(t) \\ y &= \sin(t) \end{aligned}$$

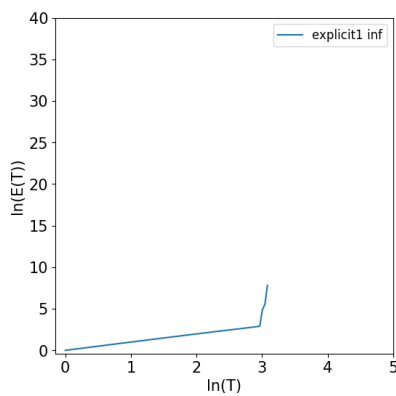
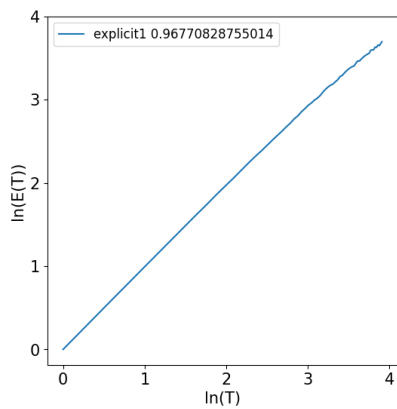
Приближений всех реализованных методов при $\tau = 0.001$:



Явный метод Эйлера 1-го порядка

При $\lambda = 10$ наклон равен 0.967:

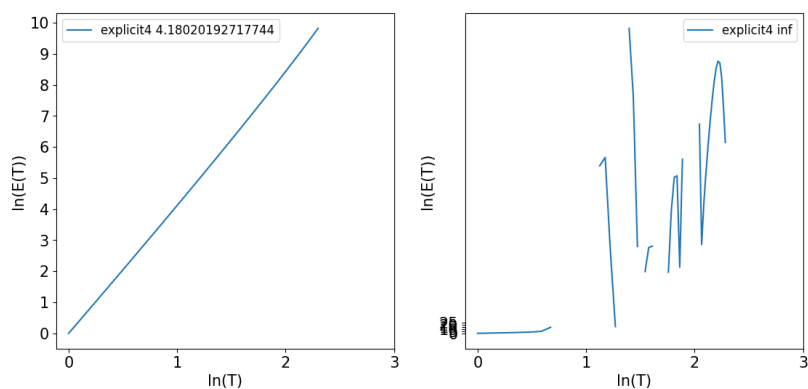
При $\lambda = 100$ наклон равен ∞ :



Метод Рунге-Кутты 4-го порядка

При $\lambda = 10$ наклон равен 4.180:

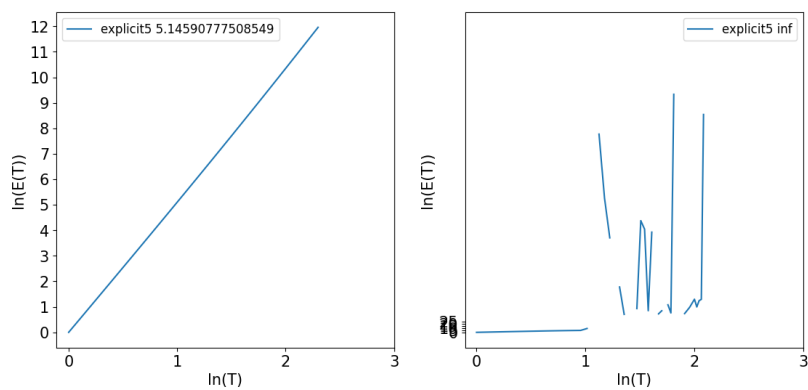
При $\lambda = 100$ наклон равен ∞ :



Метод Рунге-Кутты-Фельберга 5-го порядка

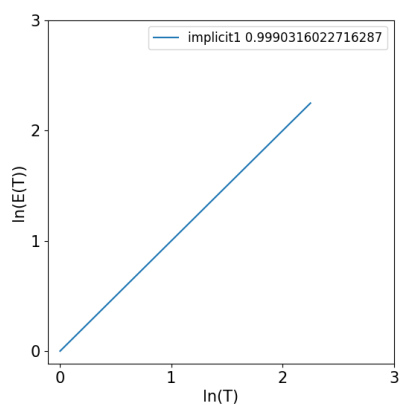
При $\lambda = 10$ наклон равен 5.145:

При $\lambda = 100$ наклон равен ∞ :



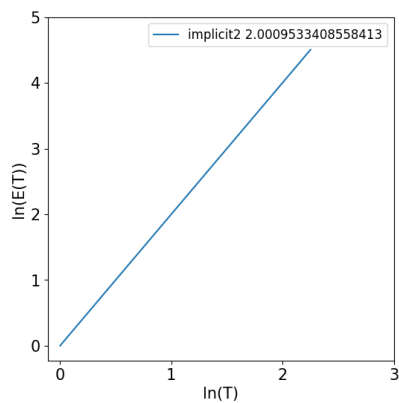
Неявный метод Эйлера 1-го порядка

При $\lambda = 10000$:



Метод Кранка-Николсона 2-го порядка

При $\lambda = 10000$:



Вывод

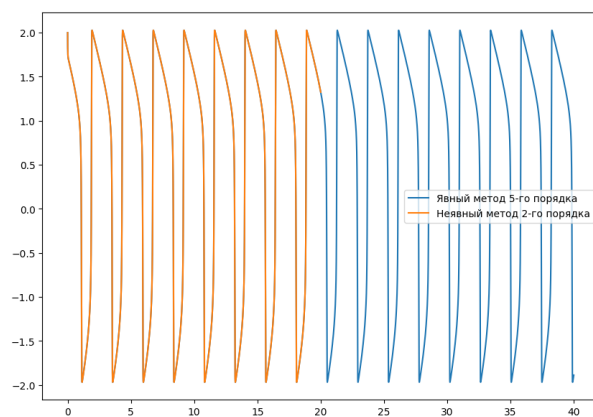
Анализ нашей задачи из-за ее жесткости лучше будет производить с помощью неявного метода, так как он устойчив даже при высокой степени жесткости задачи.

Применение методов к поставленной задаче, анализ результатов

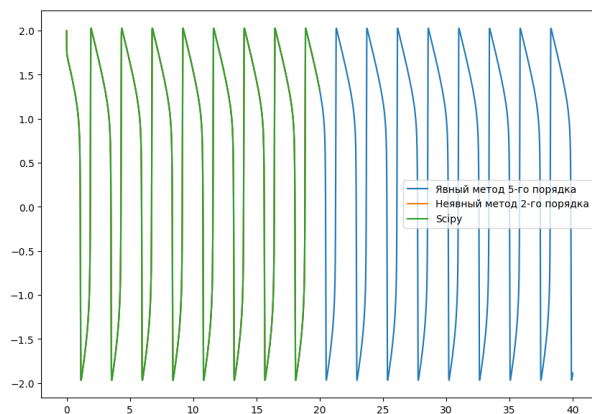
Исследование уравнения *Бонгофферра Ван дер Поля* проведем с помощью явной схемы 5-го порядка и неявной 2-го.

Проверка решения на корректность

Сравним график $x(t)$, полученный методами явного 5-го порядка и неявного 2-го — они идеально совпадают.



В качестве проверки решения приведем графики полученные с помощью методов численного интегрирования модуля SciPy.

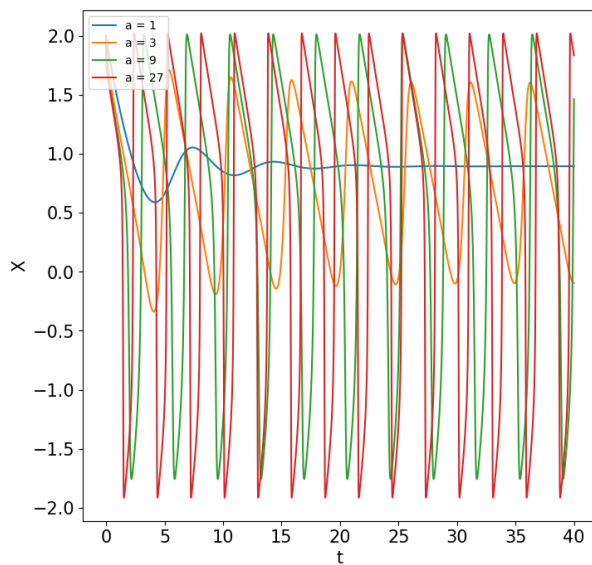


Теперь можно считать, что решение верное с точностью до погрешности методов.

Исследование уравнения в зависимости от a

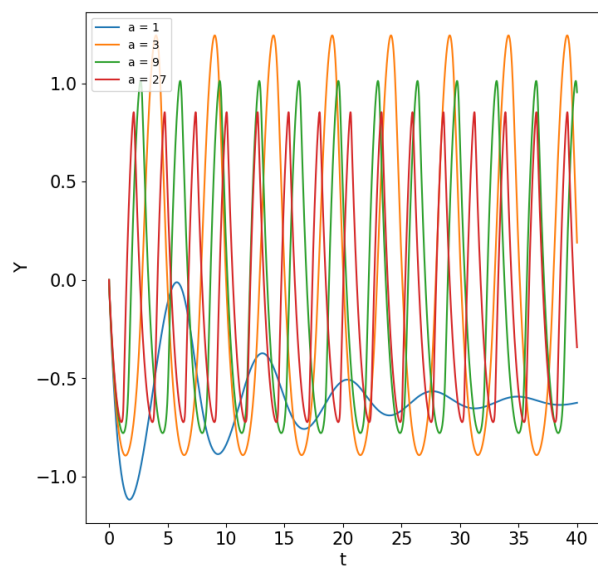
При наборе $b = 0.6, c = 0.4$

В координатах $x(t), t$

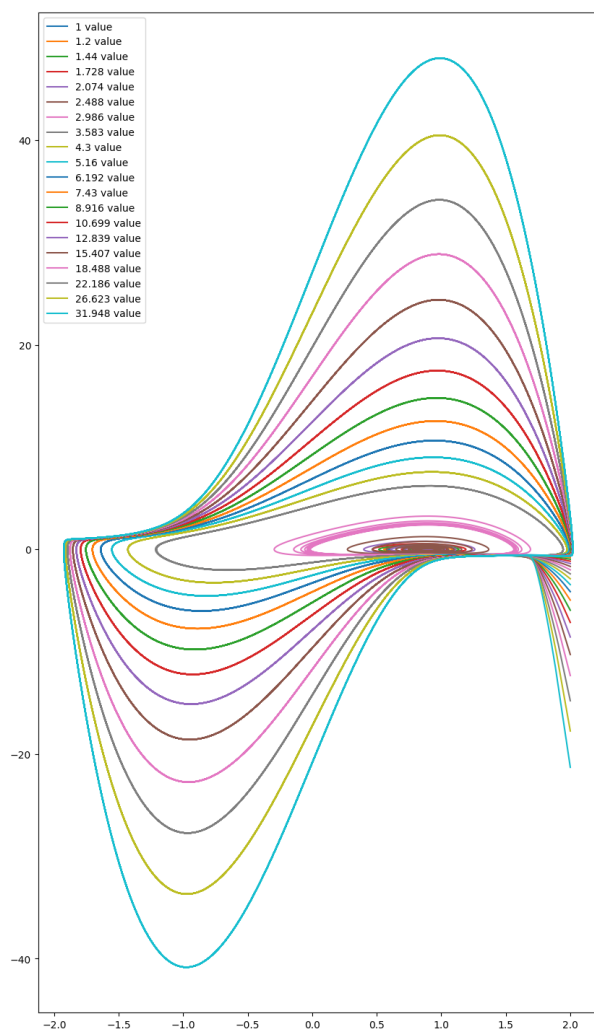


Видно, что при малых a (в зависимости от параметров b и c) происходит затухание колебаний. В то время, как при больших a периодичность $x(t)$ сохраняется и амплитуда колебаний медленно растет.

В координатах $y(t), t$ ситуация похожая, только не наблюдается рост амплитуды колебаний.

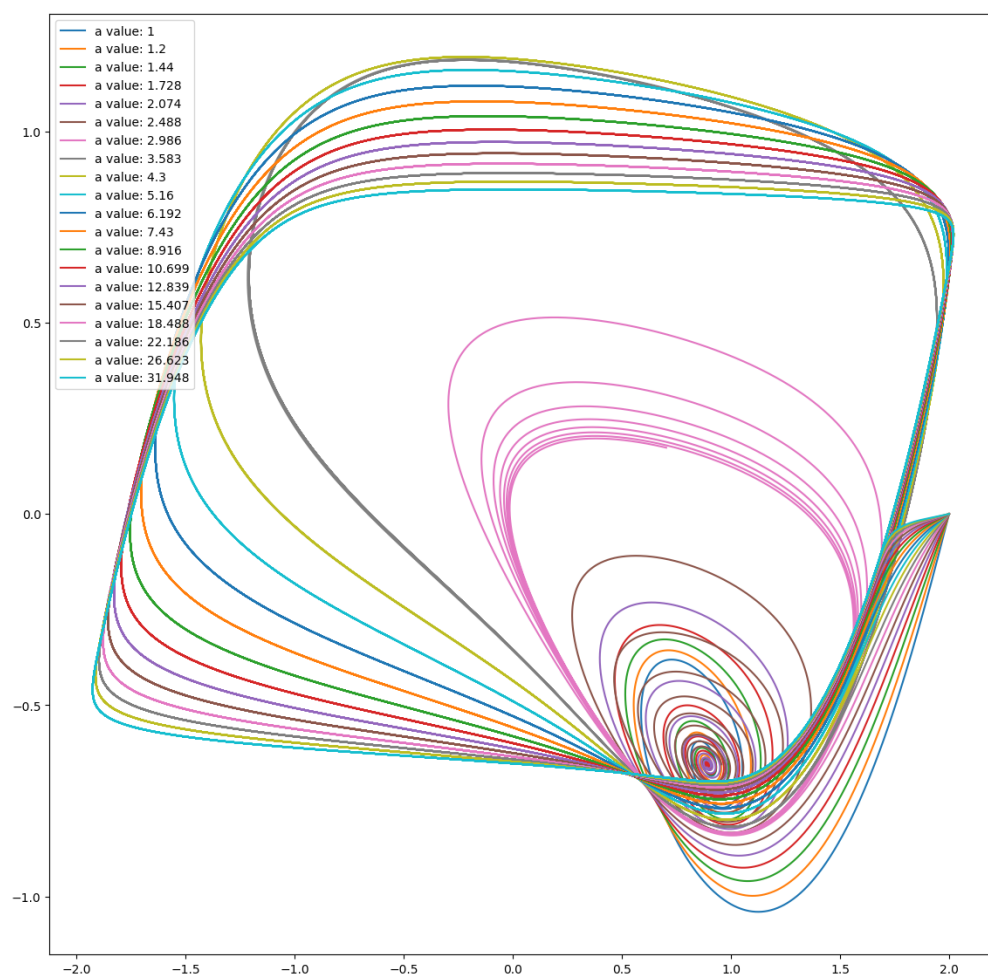


В координатах $x'(t)$, $x(t)$



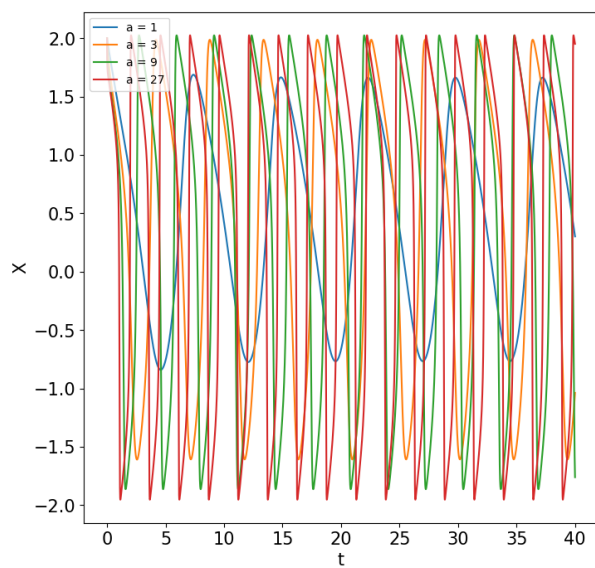
Система хаотическая при малых a и периодическая при больших.

В координатах $x(t), y(t)$ то же самое.

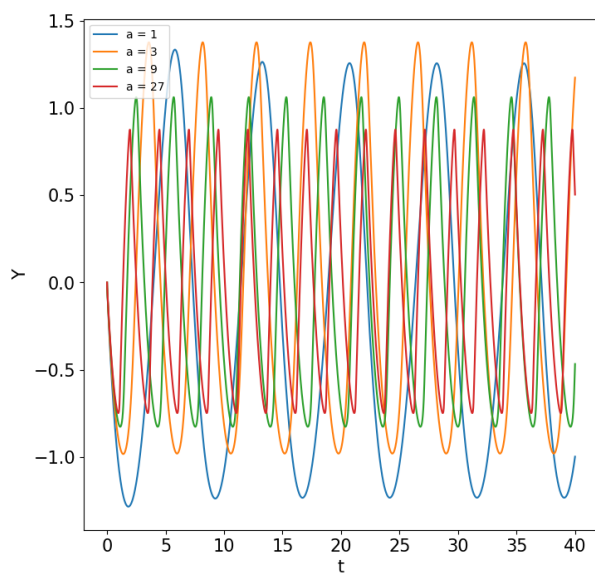


При наборе $b = 0.4, c = 0.5$

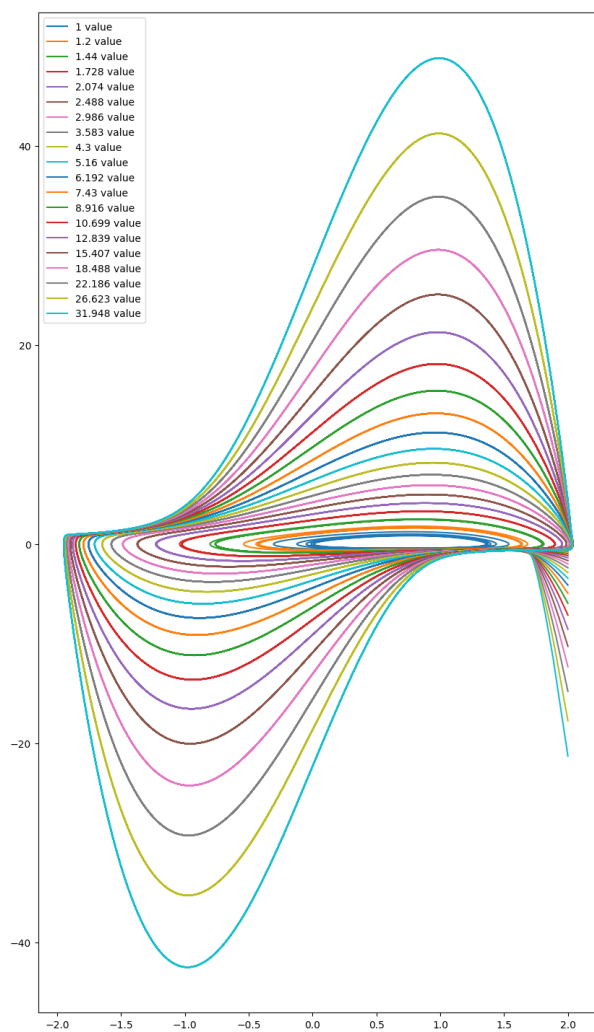
В координатах $x(t), t$



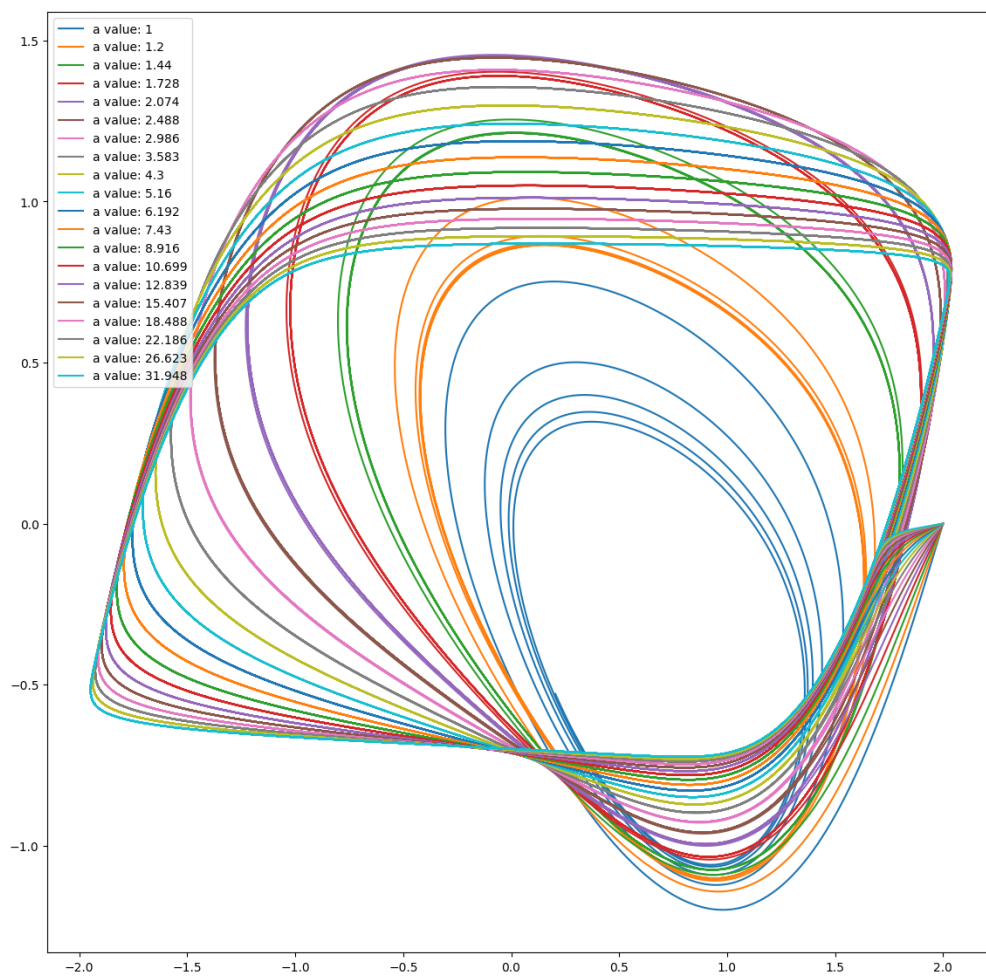
Видно, что при другом наборе параметров, изменение a влияет на систему сильнее, уже при $a = 3$ функция становится периодической.



В координатах $x'(t)$, $x(t)$



В координатах $x(t), y(t)$



При данном наборе ни при каких значениях a колебания не затухают и система периодическая.