САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И МНОГОПРОЦЕССОРНЫХСИСТЕМ

Вальков Степан Дмитриевич

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТАБАКАЛАВРА

Параметрическое исследование динамических систем с использованием систем компьютерной алгебры

НАПРАВЛЕНИЕ 010300

Фундаментальная информатика и информационные технологии

Научный руководитель
докт. физ.-мат. наук,
профессор
Андрианов С.Н.

САНКТ-ПЕТЕРБУРГ
2016

Введение

Современная наука стремится не только описать поведение изучаемого объекта, но и научиться предсказывать изменения, которые могут с ним произойти с течением времени. Конечная цель состоит в отыскании некоторого закона, позволяющего по ограниченным входным данным (например, о начальном состоянии объекта и внешним воздействиям) с точностью установить будущее объекта в любой момент времени.

Объекты и процессы, для которых однозначно определено понятие состояния – совокупности некоторых величин в конкретный момент времени, и закон, описывающий их изменение во времени, называются динамическими системами. Такие объекты и процессы встречаются в самых разных областях научного знания: в механикеи других разделах физики, в биологии и химии, в сфере вычислительных процессов и обработки информации и многих, многих других [1, 2, 6].

Эволюция динамической системы может быть описана с использованием самых разных математических моделей. Одним из наиболее применимых средств для решения этой задачи являются системы дифференциальных уравнений. Поскольку единого метода получения точного решения систем дифференциальных уравнений не существует, чаще всего прибегают к помощи численных методов решения таких систем.Однако, для разных задач требуются различные методы. Известно, что численные методы, хотя и позволяют решать большинство задач данного вида, но всё же далеки от идеала. Каждый из таких методов обладает методической погрешностью, на которую влиять без изменения метода невозможно, но и это ещё не всё.

Важной проблемой при использовании численных методов решения становится выбор правильного метода. Дело в том, что некоторые совершенно не подходят для определённых задач (например, т.н. «жёсткие»

задачи не могут быть решены простыми явными методами численного интегрирования).[10, 11]

Кроме того, современное исследование динамической системы может включать в себя работу с параметрами. Зачастую, исследователю необходимо не просто решить какую-то конкретную систему дифференциальных уравнений, а изучить целое влияние некоторых параметров на её решение, многократно получая решения для различных значений параметров.

Постановка задачи

Целью данной выпускной квалификационной работы является упрощение проведения параметрического исследования динамической системы с использованием различных методов интегрирования. Для достижения этой цели необходимо решить следующие задачи:

1) изучить основные понятий в области динамических систем;
2) изучить методы решения дифференциальных уравнений, моделирующих динамические системы;
3) разработать интерфейс для проведения параметрического исследования динамической системы в математическом пакете Maple 17;
4) провести параметрическое исследование системы с помощью созданного интерфейса на примере осциллятора Ван дер Поля.

Глава 1. Введение в предметную область

В данной главе будут кратко изложены основные положения и понятия, используемые в работе.

1.1. Динамические системы

Определение 1. Динамическая система. Динамической называют систему, состояние которой в любой момент времени после начального *($t>t_0$)*можно определить имея только начальное состояние и закон эволюции[2].

Закон эволюции можно представить с помощью дифференциальных уравнений, дискретных отображений, теории графов или цепей Маркова. В операторной форме эволюция представляется следующим образом:

$$x(t) = T_t x(t_0),$$

где T_t – оператор эволюции, а $x(t)$и $x(t_0)$ – параметры системы в моменты t и t_0 соответственно.Состояние системы в момент t получается путём применения этого оператора к начальному состоянию системы. Можно считать, что данный оператор отображает фазовое пространство системы (то есть пространство всех её состояний) само в себя.

Определение 2. Фазовая траектория.Будем считать, что состояние системы задаётся N параметрами. Тогда эволюцию системы из одного состояния в другое можно будет представить, как движение точки по некоторой траектории в N-мерном фазовом пространстве. Это и есть фазовая траектория[2].

Поскольку в данной работе важное место занимает исследование динамических систем, следует упомянуть ещё одно важное средство представления решений таких систем – ***фазовый портрет***.

Определение 3. Фазовый портрет.Если для описания динамической системы используется дифференциальное уравнение второго порядка (или

система из двух уравнений), графически их решения изображают откладывая зависимые переменные по осям координат и пользуясь независимой (чаще всего, это время) в качестве параметра. Такое изображение и называется фазовым портретом.

Обычно фазовые портреты более трудоёмки в построении, чем простые графики кривых решений, однако они же и более полезны, так как могут содержать в себе дополнительную информацию о системе. В частности, построение фазовых портретов позволяет определять особые точки системы, области, в которых система ведёт себя определённым образом и границы между такими областями [3, 12].

В качестве примера рассмотрим фазовый портрет для простого уравнения, описывающего классический маятник (оператор $\dot{}$ здесь и далееобозначает производную по времени) [2]:

$$\ddot{x} + \sin(x) = 0. \quad (1)$$

Обозначая $x_1 = x, x_2 = \dot{x}$, получим систему:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -sin(x_1) \end{cases}. \quad (2)$$

На Рис. 1 приведён фазовый портрет данной системы. Можно заметить, поведение системы вблизи точек [0±2πn,0], n = 0, 1, 2…значительно совершенно не похоже на поведение при удалении от горизонтальной оси координат. Это ***особые точки типа «центр»***, и вокруг них фазовая плоскость заполняется замкнутыми траекториями [6]. Точки [π±2πn,0] принадлежат к неустойчивым типа ***«седло»***. Интегральные кривые, проходящие через них называются ***сепаратрисами седла*** [2]. Они-то и разделяют фазовое пространство на области с совершенно различным поведением. Нетрудно заметить, что интегральные кривые вне сепаратрис уже не являются замкнутыми. Сама по себе подобная информация не кажется ценной, однако, вспоминая физическую интерпретацию рассматриваемой

системы – маятник – мы можем понять, что замкнутые интегральные кривые соответствуют колебательному движению маятника, в то время как кривые вне сепаратрис – вращательному. Таким образом, используя фазовый портрет

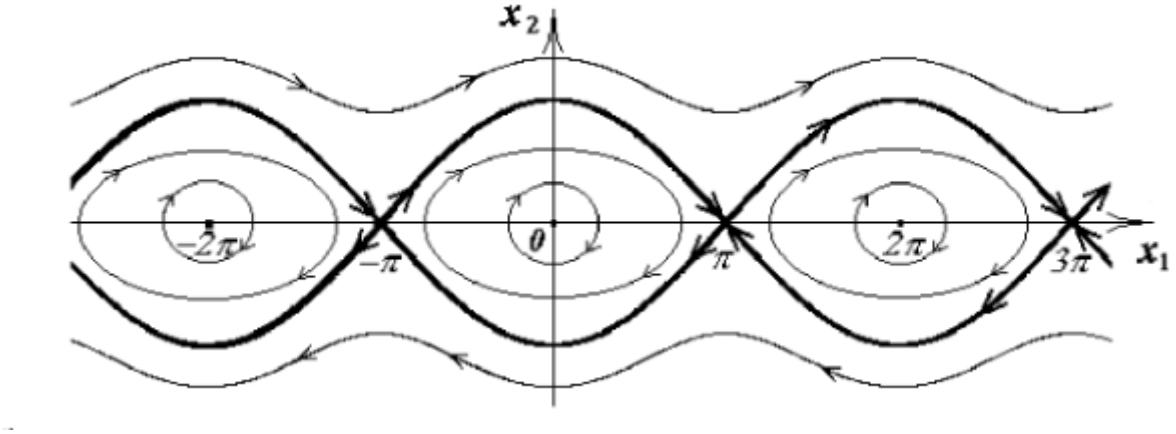


Рис. 1. Фазовый портрет для системы маятника

и задавая различные начальные значения для x_1 и x_2, (их можно трактовать, как отклонение маятника от положения равновесия и его скорость соответственно), мы можем без труда определить, какое реальное движение будет соответствовать этому начальному положению.

1.2. Численные методы решения дифференциальных уравнений

Отыскание решения дифференциального уравнения возможно различными способами. Методы решения уравнений делятся на точные, приближённые и численные. Первый класс методов даёт решение, выразимое через элементарные функции. Несмотря на то, что с аналитической точки зрения, работа с такими решениями представляется наиболее простой, их получение зачастую остаётся невозможным, потому от их использования в данной работе решено отказаться. Второй класс – приближённые методы имеют своим результатом последовательность функций, сходящуюся к точному решению [5,7,8]. Это не всегда удобно в анализе, поэтому мы сосредоточимся на численных методах решения дифференциальных уравнений.

1.2.1. Метод рядов Тейлора

Будем изучать методы, полагая, что исследуемое дифференциальное уравнение имеет вид:

$$y' = f(x, y). \qquad (3)$$

Одним из простейших с теоретической точки зрения является метод рядов Тейлора. Рассмотрим разложение функции *y(x)* в ряд Тейлора в окрестности точки x_m:

$$y(x) = y(x_m) + y'(x_m)(x - x_m) + \frac{y''(x_m)}{2!}(x - x_m) + \cdots$$

Предполагая известными решения уравнения в точках $x_0, x_1 \ldots x_m$ и считая их расположенными на расстоянии h друг от друга, мы можем получить приближенное решение в следующей точке:

$$y(x_{m+1}) = y(x_m) + hy'(x_m) + \frac{h^2}{2!}y''(x_m) + O(h^3). \qquad (4).$$

Ограничение ряда Тейлора необходимо для получения конечного решения. Тем не менее, чтобы получить решение нам всё ещё необходимо знать значения производных, а они могут быть и неизвестны в задаче. Производная первого порядка имеет выражение, непосредственно получаемое из (3):

$$y'(x_m) = f(x_m, y_m).$$

Однако уже для вычисления второй производной потребуется дополнительно посчитать частные производные для правой части уравнения (3):

$$y''(x_m) = f'_x(x_m, y_m) + f(x_m, y_m)f'_y(x_m, y_m).$$

Такая необходимость - серьёзный недостаток данного метода, поскольку для увеличения точности полученного решения, нужно увеличивать и число слагаемых в разложении (4), но с этим существенно возрастает количество необходимых вычислений, и, следовательно, время

работы любой компьютерной реализации данного метода. Тем не менее, метод является одношаговым, а потому для получения нового значения искомой функции требуется знать лишь одно предыдущее. Ещё одно преимущество – отсутствие жёсткой границы точности. Метод позволяет использовать любое количество слагаемых ряда Тейлора, тем самым получая сколь угодно близкое к точному решение, хоть и делается это ценой производительности.

1.2.2. Методы Рунге-Кутты

В этом пункте используются материалы из [5, 7, 8]. Методы Рунге-Кутты организуют последовательные вычисления по некоторой схеме. Например, метод второго порядка использует следующую схему:

$$k_1 = f(x_m, y_m),$$

$$k_2 = f(x_m + \frac{h}{2\lambda}, y_m + \frac{h}{2\lambda}k_1),$$

$$y_{m+1} = y_m + h\left((1-\lambda)k_1 + \lambda k_2\right).$$

Коэффициент λ выбирается произвольно из промежутка (0, 1]. Наиболее часто применяют данный метод с $\lambda=1/2$. На рисунке 2 приведена графическая интерпретация метода Рунге-Кутты второго порядка. Через точку (x_m, y_m) проводится касательная к интегральной кривой $y = y(x)$ - она обозначена L_1, следом определяется её угловой коэффициент k_1. Затем определяется новая точка $(x_m + h, y_m + hk_1)$, процедура повторяется для неё, а новое значение функции y_{m+1} определяется как пересечение вертикальной прямой $x = x_{m+1} = x_m + h$ и касательной с усреднённым коэффициентом. Полученное значение функции нас и интересует. Легко заметить, что для получения нового значения функции используется всего одно предыдущее, значит данный метод также является одношаговым.

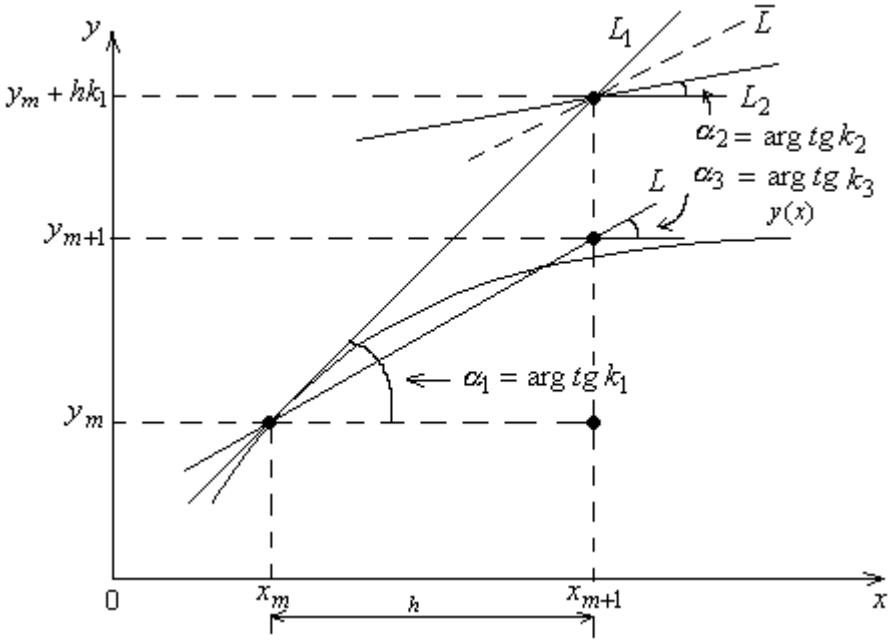


Рис. 2. Графическая интерпретация метода Рунге-Кутты второго порядка

Этот метод гораздо экономичнее метода, рассмотренного в предыдущем пункте. Для определения следующей точки методом Рунге-Кутты второго порядка нам требуется дважды вычислить значение $f(x,y)$, что значительно проще расчётов значений трёх функций в методе рядов Тейлора. Кроме того, нам не требуется отыскивать производные функции $f(x,y)$, что само по себе значительно ускоряет работу данного метода по сравнению с предыдущим.

Рассмотренный метод второго порядка не является самым употребимым методом семейства Рунге-Кутты. Это место занимает несколько более сложный, но гораздо более точный метод четвёртого порядка. Приведём формулы для него[5]:

$$k_1 = f(x_m, y_m),$$

$$k_2 = f(x_m + \frac{h}{2}, y_m + \frac{h}{2}k_1),$$

$$k_3 = f(x_m + \frac{h}{2}, y_m + \frac{h}{2}k_2),$$

$$k_4 = f(x_m + h, y_m + hk_3),$$

$$y_{m+1} = y_m + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

Основным методом, используемым в данной работе станет метод Рунге-Кутты четвёртого порядка, но в качестве дополнительного будет применён метод рядов Тейлора пятого порядка.

1.3. Осциллятор Ван дер Поля

Осциллятор Ван дер Поля является одним из классических примеров двухпараметрической колебательной системы [2]. Он относится к автоколебательным системам, то есть системам, в которых существует возможность периодического ассимптотически устойчивого движения.

Траектории движения в фазовом пространстве для таких систем представляют собой замкнутый контур, называемый предельным циклом, к которому вне зависимости от начальных условий «притягиваются» траектории из некоторой окресности. Этим система и интересна: изменяя её параметры можно получать различные формы предельного цикла и регулировать скорость движения траекторий к циклу.

Уравнение движения такого осциллятора выглядит следующим образом[2]:

$$\ddot{x} - a(1 - bx^2)\dot{x} + x = 0.$$

Перепишем его в фазовых координатах, получая систему двух дифференциальных уравнений первого порядка:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a(1 - bx_1^2)x_2 - x_1 \end{cases}.$$

Данная форма подойдёт для дальнейшего исследования, в ходе которого мы постараемся выяснить влияние каждого из параметров на поведение системы.

Глава 2. Проведение исследования

2.1 Реализация программного продукта

В данном параграфе мы рассмотрим требования, которым должен удовлетворять итоговый продукт, его функциональность; определим систему, позволяющую реализовать необходимую функциональность; опишем архитектуру приложения.

2.1.1. Требования к программному продукту

Как было описано во введении к данной работе, динамические системы встречаются в самых разных областях научного знания. Именно поэтому одним из основных требований к конечному продукту становится универсальность, в смысле независимости от предметной области. Конечный продукт должен позволять пользователю, интересующемуся любой предметной областью, решать задачу параметрического исследования динамической системы без серьёзных трудностей. Данное требование можно выполнить двумя способами:

1) разрешение использования любых условных обозначений для переменных и параметров, входящих в решаемую задачу;
2) использование строгой системы обозначений для решаемой задачи.

Первый способ представляется максимально удобным для пользователя, однако он сопряжён с неизбежными трудностями в программировании. Дело в том, все популярные математические пакеты обычно требуют явного указания, какие символы, входящие в выражение являются независимыми переменными, какие – функциями и какие – параметрами. В таких условиях понадобится либо реализовать дополнительный ввод от пользователя, где он явно укажет, какие символы в его задаче отвечают за какие математические структуры, либо охватить максимально широкий круг предметных областей и «научить» программный продукт определять по вводимой задаче, к какой области принадлежит эта

конкретная задача и интерпретировать ввод соответствующим образом. Если первый вариант ещё представляется реализуемым, хотя и не самым удобным, то его альтернатива попросту невозможна в реализации – никогда нельзя гарантировать, что охвачены абсолютно все предметные области с их характерными обозначениями, не говоря уже об очевидных трудностях, связанных с использованием разными учёными одних и тех же символов в разных значениях.

Поэтому в данной работе решено остановиться на использовании универсальной нотации.Данный подход достаточно прост в реализации, при этом понятен любому человеку, знакомому с дифференциальными уравнениями. Используемая нотация описана в ПриложенииА.

Следующие требования можно объединить в группу «функциональных».

Их содержание очевидно:

- программный продукт должен позволять пользователю решать задачи Коши для систем из двух дифференциальных уравнений и для дифференциальных уравнений первого порядка;
- пользователь должен иметь возможность использовать три параметра в своей системе;
- в продукте необходимо предусмотреть возможность получения значения решения в конкретной точке;
- для систем уравнений требуется построение фазового портрета, а для отдельных уравнений необходимо предусмотреть построение фазовых траекторий.

Кроме того, как было упомянуто в главе 1 данной работы, планируется использование методов Рунге-Кутты четвёртого порядка и рядов Тейлора

пятого, чтобы предоставить пользователю возможность сравнить скорость и точность данных методов.

В отличие от требования универсальности, данная группа требований не требует каких-то дополнительных пояснений. Этот список скорее является ориентиром для фактической реализации продукта и определяет архитектуру приложения.

2.1.2. Выбор математического пакета

Для реализации программы, удовлетворяющей описанным в предыдущем параграфе требованиям понадобится мощный математический пакет, включающий в себя инструментарий для работы с дифференциальными уравнениями. Несмотря на кажущуюся сложность задачи, таких пакетов достаточно много. В данной работе принято решение остановиться на пакете Maple 17. Для этого есть несколько причин:

- данный пакет имеет встроенные реализации для численных методов, предполагаемых к использованию вработе;
- Maple 17 имеет встроенный инструмент для создания графического интерфейса (Maplets);
- язык, используемый для программирования в Maple имеет простой и интуитивно понятный синтаксис, что значительно упрощает программирование и понимание кода, даже для начинающего пользователя;
- Maple предоставляет возможность для работы с популярными языками программирования (например, C++ и Java)[13], что не будет использоваться в данной работе напрямую, но может быть полезно при развитии приложения в будущем.

Альтернативные пакеты (например, Mathematica), хотя и схожи по функциональности, но всё же имеют свой синтаксис, который порой

вызывает трудности, как при написании программ, так и при изучении уже готового кода.

2.1.3. Архитектура приложения

В этом пункте будет приведено описание используемых функции и логики работы программы. С кодом программы можно ознакомиться в ПриложенииС.

Программа, разработанное в рамках данной работы можно разделить на два блока: функциональная часть и интерфейс.

В основе **первого блока** – инструмент ***dsolve***, позволяющий находить решения дифференциальных уравнений и систем дифференциальных уравнений как в символьном виде, так и численно. Этот инструмент включает в себя реализации самых разных методов численного решения дифференциальных уравнений, в том числе и методы Рунге-Кутты и рядов Тейлора. Результат работы этого инструмента представляет собой множество состояний системы, задаваемой ему в качестве аргумента. Кроме того он поддерживает и добавление параметров в системы дифференциальных уравнений, что является ключевым пунктом в данной работе.

Важный вспомогательный инструмент – содержащаяся в пакете ***plots*** процедура ***odeplot***.Она позволяет строить графики решений дифференциальных уравнений, полученных численно. На Рис. 3 приведён пример использования данных функций для системы негармонического осциллятора, описываемого соотношениями [9]:

$$\begin{cases} \dot{q} = p \\ \dot{p} = -q + \alpha\dfrac{q^3}{6} \end{cases}$$

В используемом фрагменте кода задаются условия задачи коши, и значение параметра, затем собирается система дифференциальных

уравнений. После этого она решается с помощью dsolve, а odeplot применяется для построения фазового портрета системы.

```
restart;
q0 := .8 : t_0 := 0 : p_0 := 0.0 : alpha := 7 :
eq := {diff(q(t), t) = p(t), diff(p(t), t) = -q(t) + alpha·q(t)^3/6, q(t_0) = q0,
    p(t_0) = p_0} :
de := dsolve(eq, type = numeric, method = taylorseries, order = 5, [q(t), p(t)]) :
with(plots) : odeplot(de, [q(t), p(t)], -100 ..100, numpoints = 1000)
```

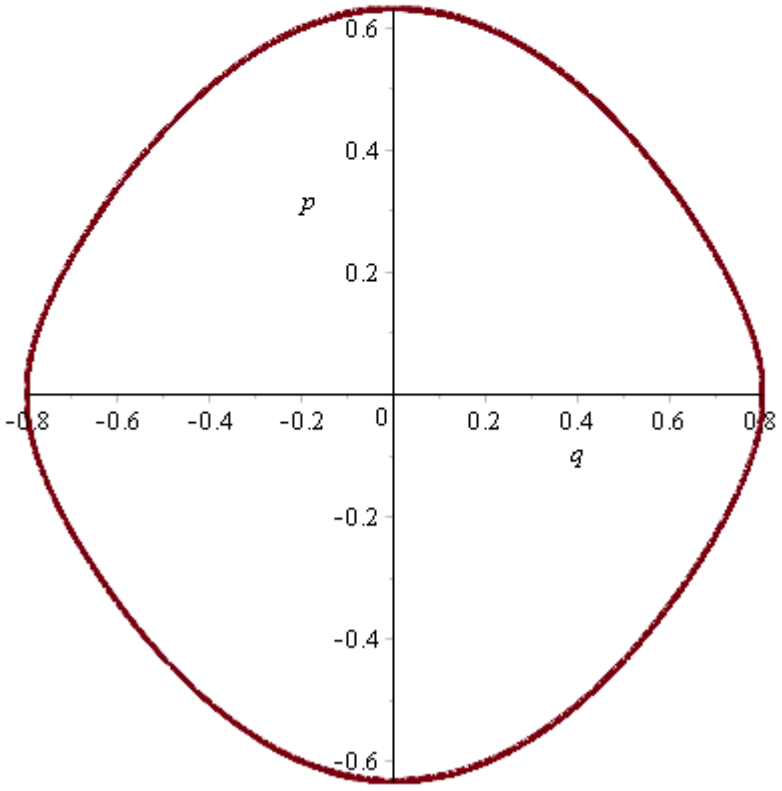


Рис. 3. Пример использования инструментов dsolve и odeplot в Maple

Второй блок реализует графический интерфейс для доступа к функциональным возможностям разработанного приложения. В его основе лежит пакет ***Maplets***, упомянутый в предыдущем пункте. Он предоставляет всё необходимое для создания оконного приложения: различные структурные элементы и способы их взаимодействия друг с другом, пользователем и функциональной частью программы.

Поскольку программный продукт предполагается к использованию только в научных исследованиях, интерфейс его организован достаточно лаконично. Верхнюю часть окна занимает область вывода графика решения или фазового портрета, в зависимости от задачи, а нижняя часть отведена под поля ввода и управляющие кнопки. Ознакомиться с видом интерфейса можнов ПриложенииВ.

Логика работы программы достаточно проста для понимания. При старте поля программы заполнены данными, описывающими систему негармонического осциллятора, упомянутую выше в обозначениях, описанных в ПриложенииА она принимает вид:

$$\begin{cases} y'(t) = x(t) \\ x'(t) = -y(t) + a\frac{x(t)^3}{6}. \end{cases}$$

Данная система используется только в качестве примера – пользователь может запустить программу с ней и ознакомиться с результатами работы программы, что-то поменять на своё усмотрение и таким образом быстрее привыкнуть к интерфейсу.

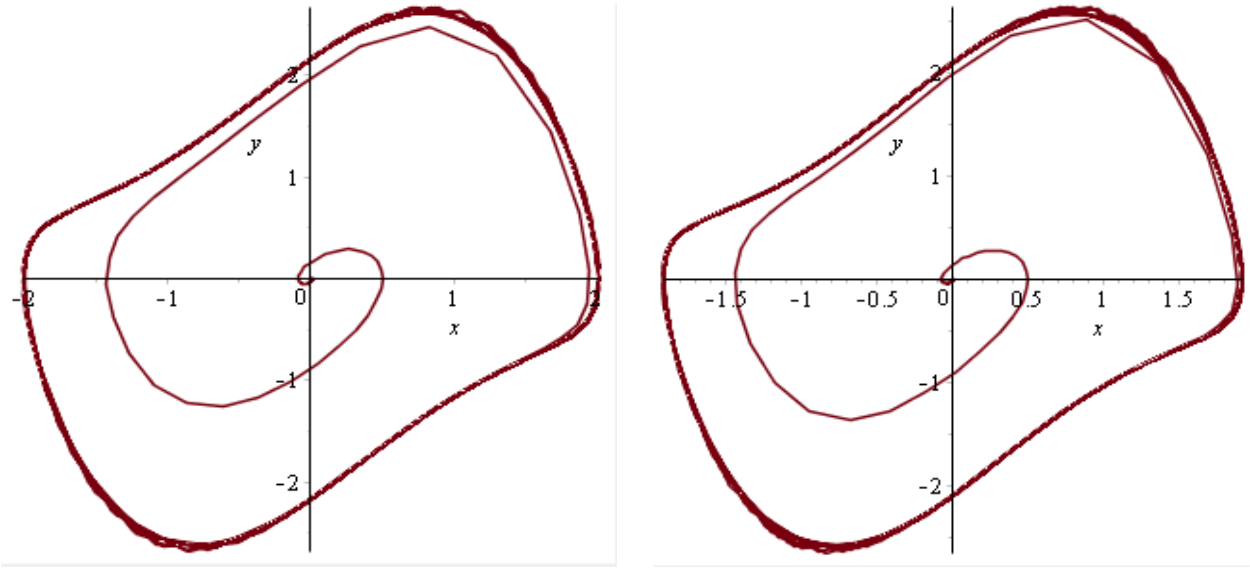
Далее предполагается, что пользователь заполнит все поля своими данными: введёт уравнения для изучения; задаст начальные условия и значения параметров, если необходимо; выберет метод и нажмёт одну из управляющих кнопок в нижней части окна. После этого все введённые пользователем данные передаются функциональной части, анализируются и исходя из них, при условии их корректности, строится решение системы уравнений или уравнения. График полученного решения (или фазовый портрет для системы) отображается в специальной области интерфейса, и при необходимости в специальном поле выводится значение решения системы в определённой точке.

2.2. Параметрическое исследование системы осциллятора Ван дер Поля

Воспользуемся разработанной программой для проведения исследования влияния параметров на поведение системы Ван дер Поля, описанной в первой главе. Для этого приведём систему к принятой в работе нотации:

$$\begin{cases} \dot{x}(t) = y(t) \\ \dot{y}(t) = -x(t) + a * y(t) * (1 - b * x(t)^2) \end{cases}.$$

Будем использовать начальные условия x(0) = 0.5 и y(0) = 0 и положим первоначальные значения параметрова = b =1.На полученном фазовом портрете отчётливо виден предельный цикл, напоминающий по форме искривлённый прямоугольник, повёрнуты относительно осей координат. К предельному циклу «притягивается» траектория, проходящая через начальную точку (0.5, 0) (Рис.4, а). Попробуем изменить параметры, не меняя их соотношения: положим a=b=1.1. Видим, что форма цикла изменилась незначительно, однако траектория, проходящая через начальную точку,

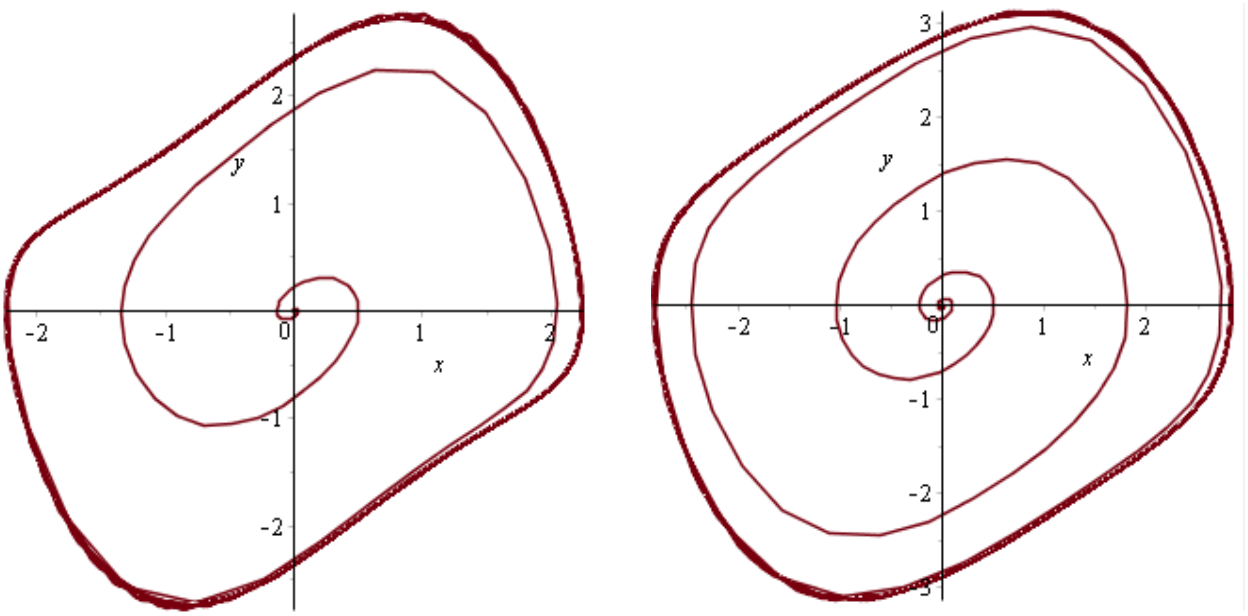


а б

Рис. 4. Фазовый портрет исследуемой системы при a=b=1 (а) и a=b=1.1 (б)

добирается до цикла гораздо быстрее (Рис. 4, б).

Исходя из результатов, мы можем выдвинуть предположение о том, что значительное влияние на структуру цикла могут оказать лишь серьёзные изменения параметров, либо изменение их соотношения. Дополнительно проверим значения параметров меньше единицы: 0.8 и 0.5 (Рис. 5).



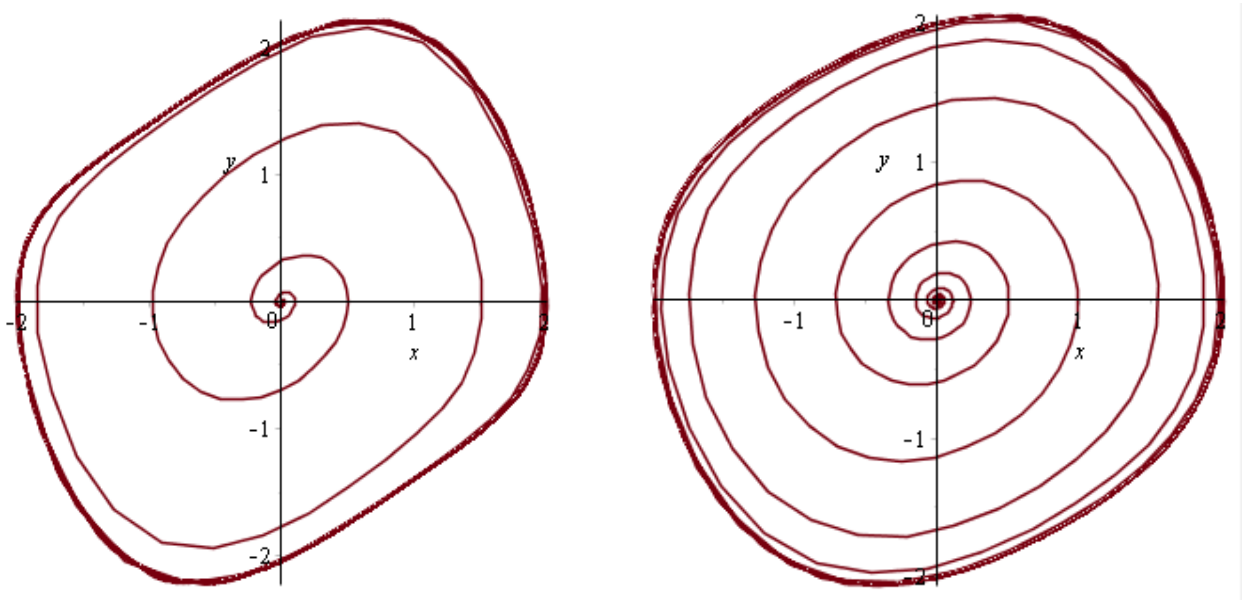
а б

Рис. 5. Фазовый портрет исследуемой системы при a=b=0.8 (а) и a=b=0.5 (б)

Нетрудно заметить, что траектория, выходящая из начальной точки делает больше и больше витков перед достижением предельного цикла с уменьшением параметров системы. В то же время, форма цикла медленно изменяется: «длинные» стороны выправляются и сам цикл становится более округлым, при этом расширяясь.

Однако до сих пор мы изменяли значения параметров синхронно. Зафиксируем значение параметра bравным единице и будем значительно изменять параметр a. Полагая a=0.5, форму цикла, очень похожую на предыдущую, но при несколько быстрее сближающейся с циклом траектории, проходящей через начальную точку. Продолжая уменьшать

параметр a при фиксированном b, можно заметить, что форма цикла всё более походит на окружность, а траектория из начальной точки делает всё

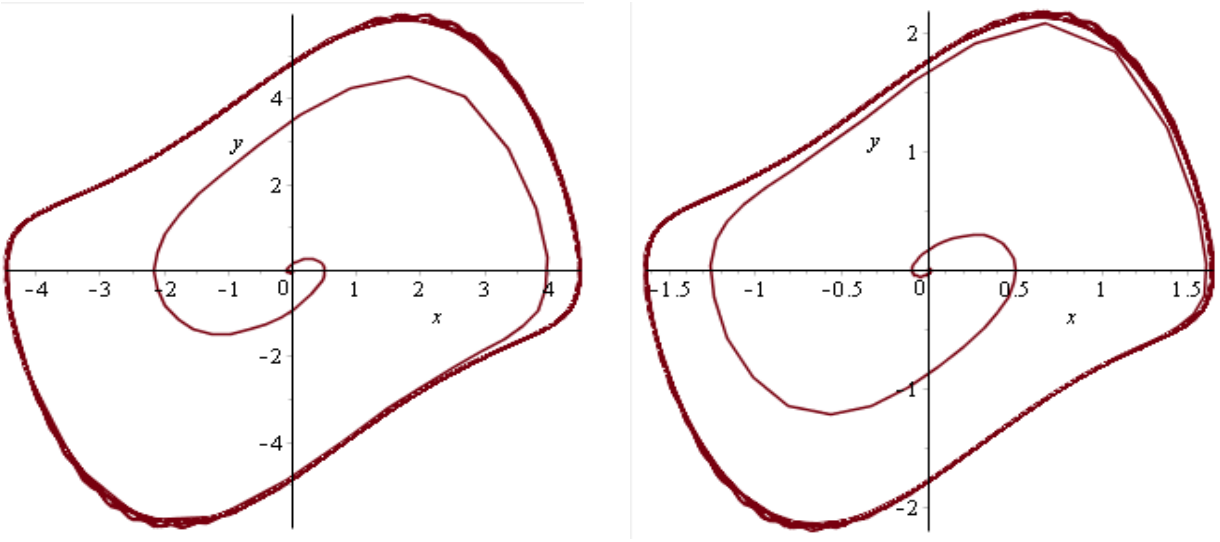


а б

Рис. 6. Фазовый портрет исследуемой системы при a=0.5, b=1 (а) и a=0.2, b=1 (б)

больше и больше витков (Рис.6). Исходя из этого можно сделать вывод, что параметр a имеет ключевое влияние на систему и определяет её структуру в значительной степени.

Зафиксируем параметр a, изменяя при этом b. В результате получим фазовые портреты, практически не отличающиеся структурой цикла, лишь с незначительными изменениями в скорости приближения траектории, проходящей через начальную точку к циклу: чем больше b, тем быстрее это происходит. В то же время, при малых значениях b значительноувеличиваютсяразмеры предельного цикла. (Рис.7). Такой результат позволяет сделать вывод об относительно слабом влиянии параметра b на качественные характеристики исследуемой системы, но заметном изменении количественных характеристик при изменении значения b.



а б

Рис. 7. Фазовый портрет исследуемой системы при a=1, b=0.2 (а) и a=1, b=1.5 (б)

Полученные результаты позволяют предсказывать поведение системы при изменении параметров, что может быть полезно при реальных исследованиях. Несмотря на то, что данная система достаточно проста, подобный подход можно использовать и для сложных систем, моделирующих реальные процессы. Информация, получаемая из параметрического исследования динамической системы таким способом может помочь в построении инфраструктуры для реально функционирующей системы, поскольку спрогнозировав изменение определённых параметров, можно с лёгкостью предсказать и изменения в системе, следующие за ними. Зная о возможных изменениях, можно подготовиться к их использованию, если они положительны, или к применению контрмер – если такие изменения нежелательны.

Заключение

Проведение параметрического исследования динамической системы потребовало изучения основные понятия области динамических систем, а так же разработки программа, позволяющая провести такое исследование для системы, представимой в виде системы двух дифференциальных уравнений с числом параметров до трёх. Проведено параметрическое исследование системы осциллятора Ван дер Поля и определено влияние параметров на эту систему. На основании полученных результатов сделан вывод о полезности проведения параметрического исследования динамической системы в реальных задачах.В дальнейшем работа может быть продолжена в различных направлениях.

1) Усложнение вводимых данных. Возможна разработка программного обеспечения с аналогичными функциями, но для более сложных динамических систем.

2) Упрощение работы с интерфейсом.

3) Реализация более сложных, но более продуктивных алгоритмов решений дифференциальных уравнений.

4) Автоматизация параметрического исследования.

Поведённое исследование позволяет вынести на защиту следующие положения:

1) изучены основные термины, касающиеся динамических систем и методы численного решения дифференциальных уравнений;

2) разработан программный продукт, позволяющий провести параметрическое исследование динамической системы;

3) проведено параметрическое исследование осциллятора Ван дер Поля, доказывающее работоспособность программы.

Список литературы

1) Андрианов С.Н. Динамическое моделирование систем управления пучками частиц. СПб.: Изд-во С.- Петерб. ун-та, 2002. 376 с.

2) Анищенко В.С. - Знакомство с нелинейной динамикой. АНО «Институт компьютерных технологий», 2002. 144 с.

3) Д. В. Аносов, Гладкие динамические системы. Гл.1. Исходные понятия, Итоги науки и техн. Сер.Соврем. пробл. мат. Фундам. направления, 1985,том 1, 156–178

4) Амелькин В.В. - Дифференциальные уравнения в приложениях. М.: Наука, 1986. 154 с.

5) Бахвалов, Н. С. - Численные методы / Н. С. Бахвалов, Н. П. Жидков, Г. М. Кобельков. – М.: Наука, 2003. 629 с.

6) Бутенин Н. В., Неймарк Ю. И., Фуфаев Н. Л. - Введение в теорию нелинейных колебаний. М.: Наука, 1987. 382 с.

7) Калиткин Н. Н. - Численные методы. М.: Наука, 1978. 500 с.

8) В.И. Мышенков, Е.В. Мышенков.- Численные методы, часть вторая. Численное решение обыкновенных дифференциальных уравнений. ИздательствоМосковскогогосударственногоуниверситеталеса, 2005. 108 с.

9) CohenD., JahnkeT., LorenzK., LubichC. NumericalIntegratorsforHighly Oscillatory // Analysis, Modeling and Simulation of Multiscale Problems, 2006. P. 553 – 576

10) Ernst Hairer, Introduction to geometric numerical integration, /Springer Series in Computational Mathematics, Vol. 31, 2002

11)Rangarajan G. Symplectic integration of nonlinear Hamiltonian systems/ Pramana – journal of physics, 1997. Vol. 48, No. 1. P. 129 – 142/

12) http://bourabai.ru/cm/le13.htm#13 // Интернет-ресурс, содержащий примеры использования различных функций Maple

13) http://www.maplesoft.com/products/Maple/// Официальный портал Maple, содержащий всю необходимую информацию о пакете

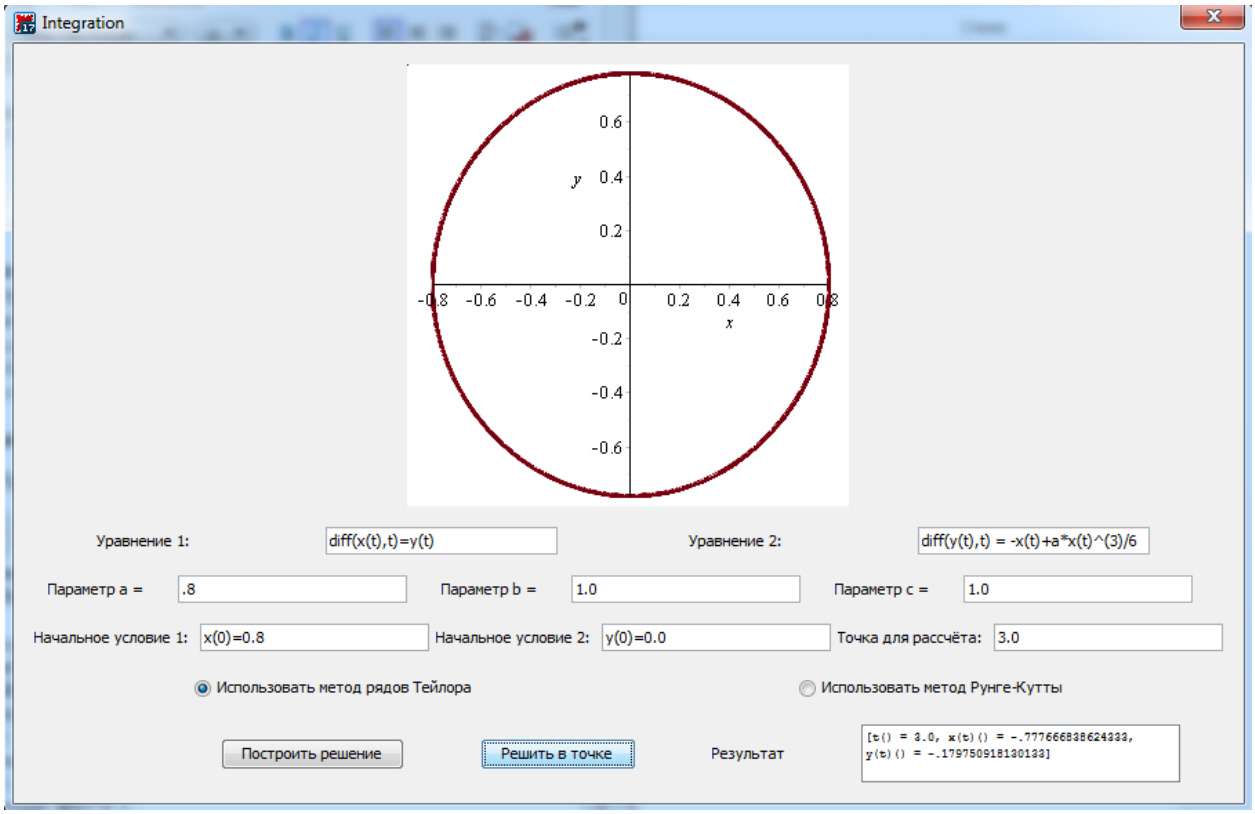
14)http://www.maplesoft.com/support/help/// Онлайн-версия документацииMaple

Приложение А. Используемая нотация

Для корректной работы разработанной требуется использование нотации, описанной в данном приложении:

- Для случая одного уравнения необходимо обозначить независимую переменную буквой t, а функцию – x(t);
- Для случая системы уравнений используются обозначения y(t) и x(t) для функций и tдля независимой переменной;
- Допускается использование трёх параметров, они должны быть обозначены латинскими буквами a, bи c;
- Необходимо соблюдать требования Maple к составлению выражений. Подробнее о них можно узнать по ссылке [14].

Приложение В. Графический интерфейс программы Integration с пояснениями



Опишем элементы графического интерфейса:

- Верхняя область. Данная часть интерфейса является зоной для отображения графиков решений или фазовых портретов системы;
- Первая строка. Здесь находятся поля ввода уравнений системы. Если пользователь желает решить одно уравнение, то оно вводится в первое поле, а второе необходимо оставить пустым;
- Вторая строка. Также содержит два поля для ввода. Здесь нужно ввести параметры a,bи c системы. Параметры должны быть числами с плавающей точкой (для целых необходимо дописать «.0»). Если система не содержит в себе параметров – данные в этих полях не используются.
- Третья строка. Здесь пользователь должен ввести начальные условия для задачи Коши. Они необходимы для построения решения. В случае

решения одного уравнения, начальное условие вводится в первое поле. Третье поле этой строки опционально – оно позволяет задать значение t, чтобы получить x(t) и y(t).

- Кнопки выбора метода. В данной работе используются два метода, поэтому пользователь может выбрать один из них. По умолчанию используется метод рядов Тейлора.

- Управляющие кнопки. Кнопка «Построить решение» запускает работу программы. После её нажатия начнётся процесс построения решения и вывода графика в верхнюю область. Перед нажатием на эту кнопку необходимо убедиться, что все поля заполнены правильно. Кнопка «Решить в точке» дополнительно отображает решение уравнения или системы в задаваемой пользователем точке. Для вывода используется соседнее текстовое поле.

Приложение C. Код программыIntegration

Данное приложение содержит код разработанной программы. Для использования достаточно вставить его в рабочий лист Maple и запустить его обработку.

```
restart;
with(Maplets[Elements]);
with(Maplets[Tools]);
with(plots);
F := proc () local a_var, b_var, eq_s, eq_s2, con, con_2, res, func, tmp, ret;
a_var := ([Get])('par_a'::float);
b_var := ([Get])('par_b'::float);
eq_s := ([Get])('eq_f'::equasion);
con := ([Get])('con_f'::equasion);
tmp := ([Get])('eq_f2');
if tmp[1] = "" then
	if ([Get])('RB1'::boolean)[1] then
		res := dsolve({parse(eq_s[1]), parse(con[1])}, type = numeric, [x(t)], method =
	classical[rk4], parameters = [a, b], maxfun = 100000);
	elif ([Get])('RB2'::boolean)[1] then
		res := dsolve({parse(eq_s[1]), parse(con[1])}, type = numeric, [x(t)], method =
	taylorseries, order = 5, parameters = [a, b]);
	end if;
	res(parameters = [a_var[1], b_var[1]]);
	func := odeplot(res, [t, x(t)], -5 .. 5, numpoints = 1000);
	([Set])('Plotter1' = func);
else eq_s2 := ([Get])('eq_f2'::equasion);
	con_2 := ([Get])('con_f2'::equasion);
	if ([Get])('RB1'::boolean)[1] then
		res := dsolve({parse(eq_s[1]), parse(eq_s2[1]), parse(con[1]), parse(con_2[1])}, type =
	numeric, [x(t), y(t)], method = classical[rk4], parameters = [a, b]);
	elif ([Get])('RB2'::boolean)[1] then
		res := dsolve({parse(eq_s[1]), parse(eq_s2[1]), parse(con[1]), parse(con_2[1])}, type =
	numeric, [x(t), y(t)], method = taylorseries, order = 5, parameters = [a, b]);
	end if;
	res(parameters = [a_var[1], b_var[1]]);
	tmp := ([Get])('target');
	if tmp[1] <> "" then
		tmp := ([Get])('target'::float);
		ret := res(tmp[1]);
	end if;
	func := odeplot(res, [x(t), y(t)], -100 .. 100, numpoints = 1000);
	([Set])('Plotter1' = func);
end if;
ret;
end proc;

with(Maplets[Elements]):
plotter := Plotter('background' = "#FFFFFF", 'continuous' = 'true', 'delay' = '100', 'height' = '220',
	'reference' = 'Plotter1', 'visible' = 'true', 'width' = '420');
```

```
c1 := "Параметр a =", TextField['par_a']('value' = .8);
c2 := "Параметр b =", TextField['par_b']('value' = 1.0);
equasion := "Уравнение 1: ", TextField['eq_f']('value' = "diff(x(t),t)=y(t)");
equasion_2 := "Уравнение 2: ", TextField['eq_f2']('value' = "diff(y(t),t) = -x(t)+a*x(t)^(3)/6");
condition := "Начальное условие 1: ", TextField['con_f']('value' = "x(0)=0.8");
condition_2 := "Начальное условие 2: ", TextField['con_f2']('value' = "y(0)=0.0");
point_ := "Точка для рассчёта: ", TextField['target']('value' = "");
ans := "Результат", TextBox['T1']('width = 40', 'height = 3', visible = true);
ans_2 := "", TextBox['T2']('width = 1', visible = false);
rk := RadioButton['RB1']("Использовать метод Рунге-Кутты", 'value' = false, 'group' = 'BG1');
tay := RadioButton['RB2']("Использовать метод рядов Тейлора", 'value' = true, 'group' = 'BG1');
meth := ButtonGroup['BG1']()
but := Button("Построитьрешение", Evaluate('T2' = "F()"));
but_2 := Button("Решитьвточке", Evaluate('T1' = "F()"));
mplt := Maplet(Window("Integration", [[plotter], [equasion, equasion_2], [c1, c2], [condition,
        condition_2, point_], [tay, rk], [ans_2, but, but_2, ans]]), meth);
Maplets[Display](mplt);
```