МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования   
**«Национальный исследовательский   
Нижегородский государственный университет им. Н.И. Лобачевского»**

**(ННГУ)**

**Институт информационных технологий, математики и механики**

**Кафедра алгебры, геометрии и дискретной математики**

**ОТЧЕТ ПО ЛАБОРАТОНОЙ РАБОТЕ**

**«Численное решение задачи Коши для ОДУ»**

**Выполнил:** студент группы 381706-2 Мышкин Андрей Александрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

**Научный руководитель:**

Старший преподаватель

Эгамов Альберт Исмаилович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Подпись

Нижний Новгород  
2020

Оглавление

[Введение 3](#_Toc37972839)

[Описание математической модели Лотки-Вольтерры 5](#_Toc37972840)

[Метод Рунге-Кутта 4-ого порядка точности 8](#_Toc37972841)

[Система разработки и структура программы 11](#_Toc37972842)

[Руководство пользователя 12](#_Toc37972843)

[Руководство программиста 16](#_Toc37972844)

[Заключение 17](#_Toc37972845)

[Литература 18](#_Toc37972846)

# Введение

Обыкновенное дифференциальное уравнение (ОДУ) — дифференциальное уравнение для функции от одной переменной. (Этим оно отличается от уравнения в частных производных, где неизвестная — функция нескольких переменных.) Таким образом, ОДУ — уравнения вида,

 (1)

где *y(x)* — неизвестная [функция](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)) (возможно, [вектор-функция](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%BA%D1%82%D0%BE%D1%80-%D1%84%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F), тогда , *F* как правило, тоже вектор-функция со значениями в пространстве той же [размерности](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B7%D0%BC%D0%B5%D1%80%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%BF%D1%80%D0%BE%D1%81%D1%82%D1%80%D0%B0%D0%BD%D1%81%D1%82%D0%B2%D0%B0); в этом случае говорят о *системе* дифференциальных уравнений), зависящая от независимой переменной *x* , штрих означает дифференцирование по *x*. Число *n* (порядок старшей производной, входящей в данное уравнение) называется порядком дифференциального уравнения.

Независимая переменная *x* часто интерпретируется (особенно в дифференциальных уравнениях, возникающих в физических и других естественно-научных задачах) как время, поэтому её часто обозначают буквой *t*. Переменная *y* — некоторая величина (или совокупность величин, если *y* является вектор-функцией), изменяющаяся со временем. Например, *y* может означать набор координат точки в пространстве; в этом случае уравнение описывает движение точки в пространстве, то есть изменение её координат с течением времени. Независимая переменная *x* обычно принимает вещественные значения, однако рассматриваются и дифференциальные уравнения, в которых переменная *x* комплексная (так называемые уравнения *с комплексным временем*).

Наиболее часто встречаются дифференциальные уравнения вида

 (2)

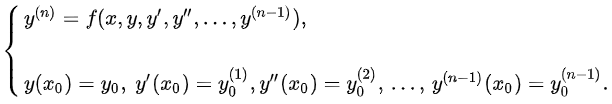
в которых старшая производная *y^(n)* выражается в виде функции от переменных *x, y* и производных *y^(i)* порядков меньше *n*. Такие дифференциальные уравнения называются нормальными или разрешёнными относительно производной.

В противоположность уравнениям вида (2), дифференциальные уравнения вида (1) называются уравнениями, не разрешёнными относительно производной или неявными дифференциальными уравнениями.

Классическим решением дифференциального уравнения (2) называется *n* раз дифференцируемая функция *y(x)*, удовлетворяющая уравнению во всех точках своей области определения. Обычно существует целое множество таких функций, и для выбора одного из них требуется наложить на него дополнительное условие. Начальным условием для уравнения (2) называется условие



где *xօ* — некоторое фиксированное значение независимой переменной (фиксированный момент времени), а *yօ* и *yօ^(i)* — соответственно, фиксированные значения функции *y* и всех её производных до порядка *n-1* включительно. Дифференциальное уравнение (2) вместе с начальным условием (3) называется начальной задачей или задачей Коши:



Теорема Пикара утверждает, что при достаточно общих ограничениях на функцию *f* , стоящую в правой части уравнения (2), задача Коши для этого уравнения имеет единственное решение, определённое на некотором интервале оси времени *x*, содержащем начальное значение *xօ* (этот интервал, вообще говоря, может не совпадать со всей осью).

Основные задачи и результаты теории дифференциальных уравнений: существование и единственность решения различных задач для ОДУ, методы решения простейших ОДУ, качественное исследование решений ОДУ без нахождения их явного вида.

# Описание математической модели Лотки-Вольтерры

*Математическая модель* — математическое представление реальности, один из вариантов модели как системы, исследование которой позволяет получать информацию о некоторой другой системе. Математическая модель предназначена предсказать поведение реального объекта, но всегда представляет собой ту или иную степень его идеализации.

*Математическим моделированием* называют как саму деятельность, так и совокупность принятых приёмов и техник построения и изучения математических моделей.

Все естественные и общественные науки, использующие математический аппарат, по сути, занимаются математическим моделированием: заменяют объект исследования его математической моделью и затем изучают последнюю. Связь математической модели с реальностью осуществляется с помощью цепочки эмпирических законов, гипотез, идеализаций и упрощений. С помощью математических методов описывается, как правило, идеальный объект или процесс, построенный на этапе содержательного моделирования.

Для данной лабораторной работы была выбрана математическая модель Лотки-Вольтерры, являющаяся системой «хищник-жертва».

**Система «хищник — жертва»** — сложная экосистема, для которой реализованы долговременные отношения между видами хищника и жертвы, типичный пример [коэволюции](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%8D%D0%B2%D0%BE%D0%BB%D1%8E%D1%86%D0%B8%D1%8F).

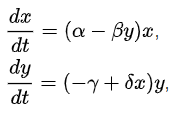
Отношения между хищниками и их жертвами развиваются циклически, являясь иллюстрацией нейтрального равновесия

**Биологическая система**

Приспособления, вырабатываемые жертвами для противодействия хищникам, способствуют выработке у хищников механизмов преодоления этих приспособлений. Длительное совместное существование хищников и жертв, приводит к формированию системы взаимодействия, при которой обе группы устойчиво сохраняются на изучаемой территории. Нарушение такой системы часто приводит к отрицательным экологическим последствиям.

Негативное влияние нарушения коэволюционных связей наблюдается при интродукции видов. В частности, козы и кролики, интродуцированные в Австралии, не имеют на этом материке эффективных механизмов регуляции численности, что приводит к разрушению природных экосистем.

В математической форме предложенная система имеет следующий вид:



где *x* — количество жертв, *y* — количество хищников, *t* — время, *α*, *β*, *γ*, *δ* — коэффициенты, отражающие взаимодействия между видами.

**Постановка задачи**

Рассматривается закрытый ареал, в котором обитают два вида — травоядные («жертвы») и хищники. Предполагается, что животные не иммигрируют и не эмигрируют, и что еды для травоядных животных имеется с избытком. Тогда уравнение изменения количества жертв (без учёта хищников) принимает вид:



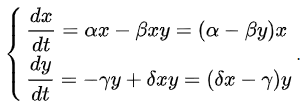
где *α* — коэффициент рождаемости жертв, *x* — величина популяции жертв *dx/dt* — скорость прироста популяции жертв.

Пока хищники не охотятся, они вымирают, следовательно, уравнение для численности хищников (без учёта численности жертв) принимает вид:



где *γ* — коэффициент убыли хищников, *y* — величина популяции хищников, *dy/dt* — скорость прироста популяции хищников.

При встречах хищников и жертв (частота которых прямо пропорциональна величине *xy*) происходит убийство жертв с коэффициентом *β*, сытые хищники способны к воспроизводству с коэффициентом *δ*. С учётом этого, система уравнений модели такова:



Для численного решения Задачи Коши для ОДУ модели Лотки-Вольтерры будет использоваться метод Рунге-Кутта 4-ого порядка точности

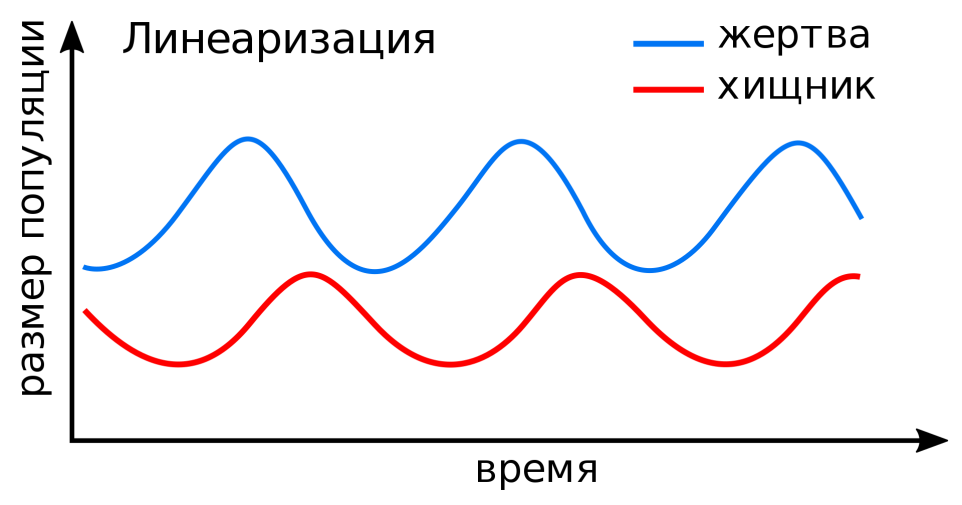


Рисунок 1. Размер популяции хищников и жертв как функция от времени в модели Лотки — Вольтерры

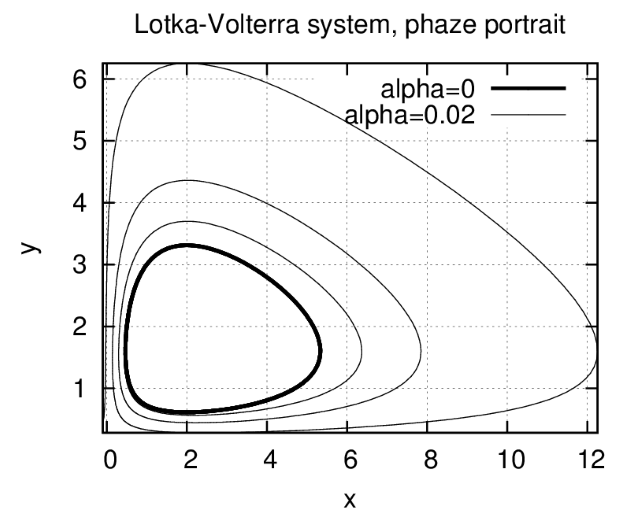


Рисунок 2. Фазовый портрет для системы Лотка-Вольтерра

# Метод Рунге-Кутта 4-ого порядка точности

Численные методы позволяют получить искомое решение y(t) дифференциального уравнения в форме таблицы его приближенных значений https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-2wOuhZ.png для заданной последовательности значений аргумента https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-_Yrf71.png.

Непрерывный отрезок https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-4z_REz.png, на котором требуется получить решение дифференциального уравнения, заменяют конечной последовательностью дискретных точек https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-LArP_K.png (узловых точек).

Величина https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-f50zN1.png называется шагом интегрирования.

Численные методы делятся на два класса: одношаговые и многошаговые.

Одношаговые методы действуют по принципу:

https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-X61iJH.png

т. е. для расчета следующего значения решения https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-_digmt.png достаточно знать только текущее значение https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-h1geYW.png. К таким методам относится и метод Рунге-Кутта.

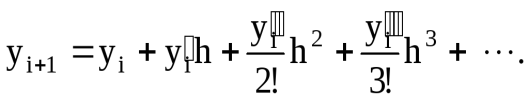
Многошаговые методы используют такую процедуру:

https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-nGLKHx.png

К ним относятся методы Адамса, Милна.

В основе одношаговых методов лежит следующая идея: искомое решение https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-GzYlZn.pngдифференциального уравнения в окрестности текущей точки https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-bbQbrn.png можно представить в виде ряда Тейлора.

Учитывая, что https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-RXs6tA.png, получим:



Производится усечение ряда Тейлора. Количество оставшихся членов ряда определяет порядок численного метода и, соответственно, его точность. При этом операция вычисления производных https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-UsmkQM.png заменяется последовательностью простейших операций над значениями функции f(t,y) в нескольких точках интервала https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-VqMgpv.png.

**Метод Рунге-Кутта**

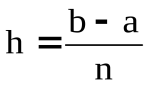
Пусть на отрезке https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-uETh_Q.png требуется найти численное решение дифференциального уравнения

https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-wE971H.png

при начальных условиях https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-vWHACv.png.

Разбиваем отрезок https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-Wq_ou9.png на https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-_JwMHr.png равных частей точками

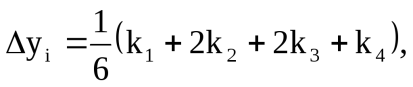
https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-0KEMh7.png,

где i = 0,1,2,3, … n;  − шаг интегрирования.

Тогда каждое последующее значение искомого решения y будет определяться так:

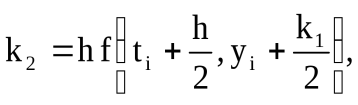
https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-h4mZQK.png,

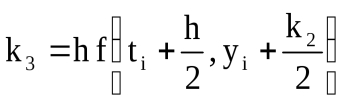
где

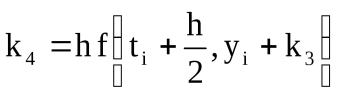


здесь

https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-EVudUQ.png



,

. (

Это метод Рунге-Кутта 4-го порядка, одношаговый, обладает достаточной точностью, его погрешность − (https://studfile.net/html/2706/189/html_NmlVsSuVin.FojS/img-ZiBxdr.png).

Метод Рунге-Кутта применяется также для решения систем дифференциальных уравнений. В этом случае от скалярной формы записи выражений переходят к векторной: y →Y, f(t,y) → F(t,Y).

Чтобы применить метод Рунге-Кутта к дифференциальному уравнению n -го порядка

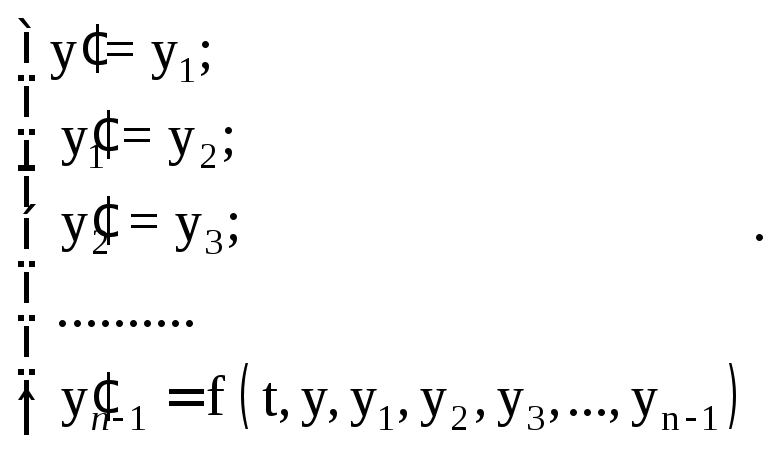
y(n) = f(t,y,y′,y″, … ,y(n-1)),

следует свести его к системе n дифференциальных уравнений 1-го порядка (в форме Коши).

Для этого вводятся обозначения для производных:

**y′=y1; y″=y2; y′′′=y3; y(n-1)=yn-1; y(n)= yn.**

Тогда результирующая система дифференциальных уравнений будет иметь вид:



# Система разработки и структура программы

Данная реализация разработана и протестирована на языке Python версии 3.8. Такой язык программирования был выбран не просто так. Это высокоуровневый язык, ориентированный на производительность и читаемость кода, также есть множество библиотек, включающих большой объем полезных функций

Программа на входе принимает данные для ОДУ модели Лотки-Вольтерры, передает их в систему дифференциальных уравнений. Далее вызывается метод Рунге-Кутта и после его завершения полученные данные отображаются на графике, постепенно строя фазовый портрет.

Все данные ползователь может ввести самостоятельно

# Руководство пользователя

При запуске программы появляется окно с графиком и необходимыми вводимыми данными, такие как: Задача Коши, коэффициенты для модели Лотки-Вольтерры, шаг и количество точек.

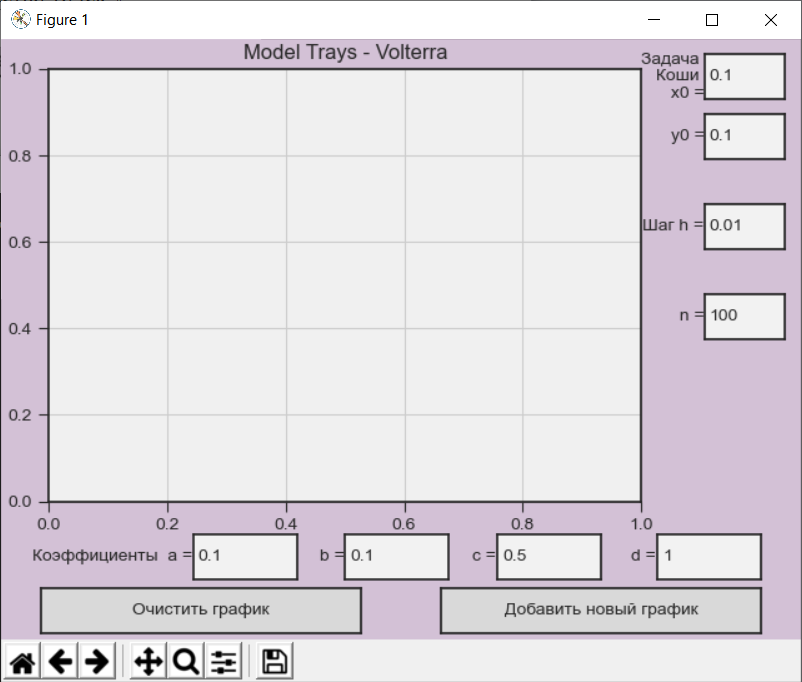


Рисунок 3

Изначально уже введены данные, которые может сразу проверить пользователь. Для построения графика нужно выбрать параметры и нажать кнопку «Добавить».

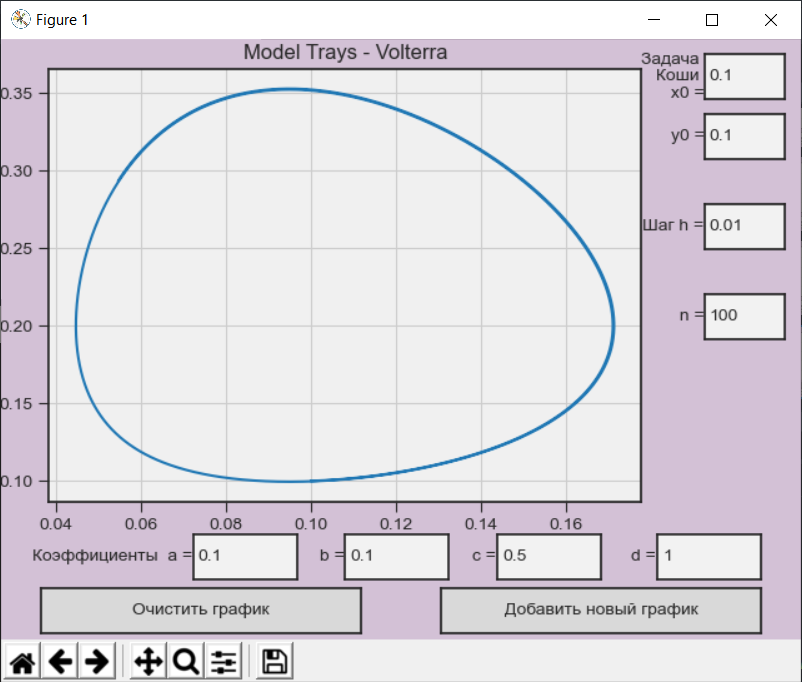


Рисунок 4

Так как множество решений, вычисленное для всевозможных начальных условий, образует фазовый портрет динамической системы, то для построения портрета необходимо нарисовать несколько графиков на одинаковых параметрах, меняя только начальные условия, то есть Задачу Коши

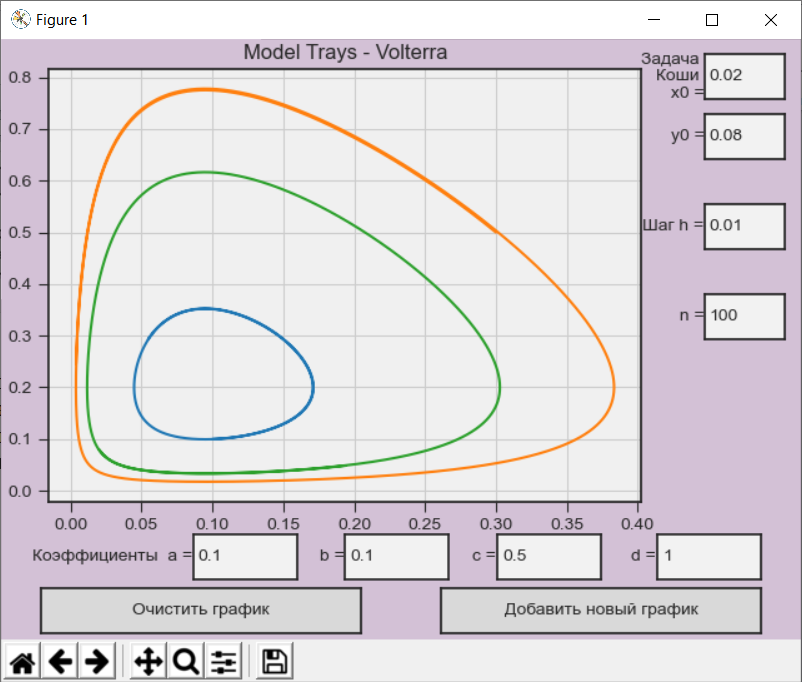


Рисунок 5

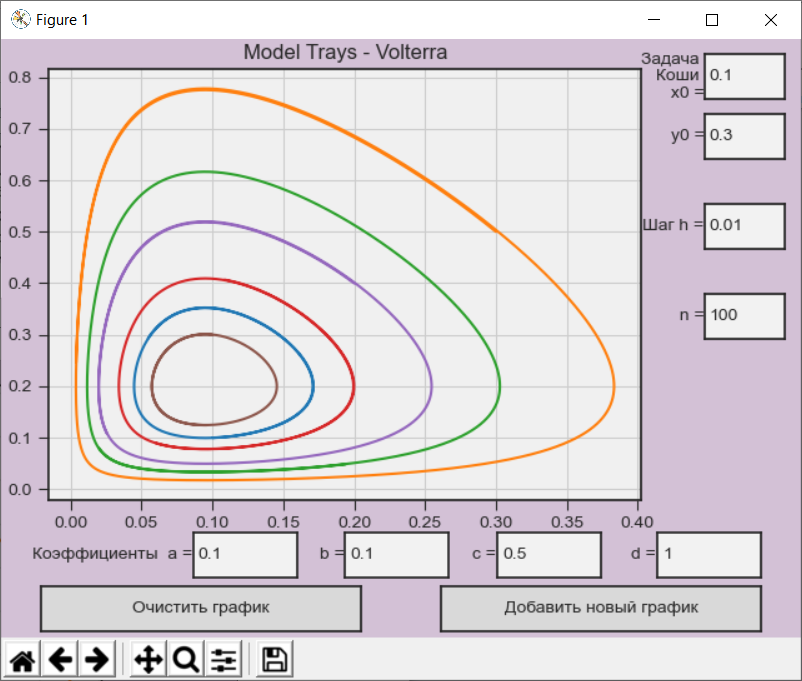


Рисунок 6

Таким образом, получаем фазовый портрет для данного ОДУ

Для изменения параметров и построения нового фазового портрета для них, пользователь может не заново запускать программу, а просто нажать на кнопку «Очистить график» и продолжать работу.

# Руководство программиста

В данной работе используется несколько функций.

def Fx\_der(x, y, a, c)

def Fy\_der(x, y, b, d) – создают систему 2 дифференциальных уравнений 1-ого порядка

def Runge\_Kutt\_4th(x0,y0,n,h) – функция метода Рунге-Кутта

# Заключение

В ходе реализации лабораторной работы были освоены методы разработки приложений на высокоуровневом языке программирования Python, в том числе и работа с графическим интерфейсом, изучены математическая модель Лотки-Вольтерры, её особенности.

Рассмотрен алгоритм построения фазового портрета ОДУ

# Литература

1. Метод Рунге-Кутта 4-ого порядка - <https://studfile.net/preview/2893465/page:3/>
2. Самарский А.А. “Введение в численный методы”, Издательство «Лань», 2005. – 228с.
3. Статья на Википедии - <https://ru.wikipedia.org/wiki/Математическая_модель#Модель_Мальтуса>
4. Статья на Википедии - [https://ru.wikipedia.org/wiki/Система\_«хищник\_—\_жертва»](https://ru.wikipedia.org/wiki/Система_)
5. Статья на Википедии - <https://ru.wikipedia.org/wiki/Модель_Лотки_—_Вольтерры>
6. Фазовые портреты динамических систем - <https://www.intuit.ru/studies/courses/3484/726/lecture/25610?page=6>