Отчет по лабораторной работе №5

по дисциплине: Математическое моделирование

Ким Михаил Алексеевич

Содержание

# 1 Цель работы

Продолжить знакомство с функционалом языка программирования Julia, дополнительных библиотек (DifferentialEquations, Plots), интерактивного блокнота Pluto, а также интерактивной командной строкой REPL. Продолжить ознакомление с языком моделирования Modelica и программным обеспечением OpenModelica. Используя эти средства, описать математическую модель Лотки-Вольтерры.

# 2 Задание

Для модели «хищник-жертва»:

Построить график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: , . Найти стационарное состояние системы.

# 3 Теоретическое введение

## 3.1 Модель «хищник-жертва» (модель Лотки-Вольтерры)

**Модель «хищник-жертва» (модель Лотки-Вольтерры)** — это математическая модель, описывающая динамику популяций двух видов, где один вид является хищником, а другой - жертвой.

В модели предполагается, что популяции хищников и жертв изменяются в зависимости от времени. Модель основывается на следующих предположениях:

1. Численность популяции жертв и хищников зависят только от времени (модель не учитывает пространственное распределение популяции на занимаемой территории).
2. В отсутствии взаимодействия численность видов изменяется по модели Мальтуса, при этом число жертв увеличивается, а число хищников падает.
3. Естественная смертность жертвы и естественная рождаемость хищника считаются несущественными.
4. Эффект насыщения численности обеих популяций не учитывается.
5. Скорость роста численности жертв уменьшается пропорционально численности хищников.

Модель Лотки-Вольтерры состоит из двух дифференциальных уравнений:

где – число жертв, - число хищников. Коэффициент описывает скорость естественного прироста числа жертв в отсутствие хищников, - естественное вымирание хищников, лишенных пищи в виде жертв. Вероятность взаимодействия жертвы и хищника считается пропорциональной как количеству жертв, так и числу самих хищников (). Каждый акт взаимодействия уменьшает популяцию жертв, но способствует увеличению популяции хищников (члены и в правой части уравнения) [1].

Модель Лотки-Вольтерры позволяет предсказать изменение численности популяций в зависимости от коэффициентов, заданных в уравнениях, и начальных условий.

Модель была разработана в 1925 году и до сих пор является одной из самых известных моделей в экологии. Она может быть применена в различных областях, таких как контроль популяции рыб, насекомых и многих других видов.

Кроме того, модель «хищник-жертва» часто используется в учебных целях для объяснения основ экологии и динамики популяций.

Важно отметить, что модель является упрощенной и не учитывает многие реальные факторы, которые влияют на динамику популяций в природе.

Однако, модель Лотки-Вольтерры имеет множество расширений и модификаций, что позволяет применять ее в более сложных ситуациях [2].

## 3.2 Стационарное состояние системы

**Стационарное состояние системы** — это состояние, при котором популяции хищников и жертв остаются почти постоянными с течением времени. В этом состоянии количество жертв и хищников остается примерно на одном уровне, так как скорости уменьшения и увеличиния популяций сбалансированы. Система может достичь стационарного состояния только в определенных условиях: при наличии достаточно большого количества жертв и хищников, а также при определенных значений коэффициентов в уравнениях модели.

Стационарное состояние системы является важным концептом в модели «хищник-жертва», так как оно позволяет предсказать устойчивость популяций в долгосрочной перспективе. Однако, в реальности популяции животных и растений часто находятся в нестационарном состоянии, поскольку в природе действуют множество факторов, которые могут влиять на их численность.

Нестационарность популяций может быть вызвана изменением климатических условий, естественными бедствиями, наличием новых хищников или болезней, действиями человека и т.д.

Поэтому в реальности модель «хищник-жертва» может быть использована только для грубой оценки тенденций изменения численности популяций в определенных условиях, в то время как реальная динамика популяций в природе может быть гораздо более сложной и нелинейной [3].

## 3.3 Малые изменения параметров в модели «хищник-жертва»

Малое изменение в модели «хищник-жертва» может привести к значительным изменениям в динамике популяций. Например, если в модель добавить еще один вид, который конкурирует за ресурсы с жертвами, это может привести к сильному снижению численности хищников и жертв.

Также изменение коэффициентов в уравнениях модели может привести к изменению динамики популяций. Например, увеличение коэффициента убийства жертв хищниками может привести к быстрому снижению численности жертв и, в конечном итоге, к снижению численности хищников.

Модель Лотки-Вольтерры может быть модифицирована для учета различных факторов, таких как миграция, конкуренция за ресурсы, изменение климатических условий и т.д. Такие дополнения позволяют более точно описывать динамику популяций в различных экологических условиях [4].

Благодаря таким модификациям, модель «хищник-жертва» может быть использована для прогнозирования динамики популяций в различных условиях, что необходимо для планирования использования биологических ресурсов и сохранения биоразнообразия.

Например, модель может быть использована для прогнозирования того, как изменения климата могут повлиять на популяции животных и растений в определенном регионе. Это может помочь при разработке стратегий адаптации объектов к изменненым условиям и стратегий охраны биоразнообразия видов. Также модель «хищник-жертва» может быть использована для определения оптимального уровня охоты на животных для сохранения их популяции.

В целом, модель Лотки-Вольтерры остается важным инструментом для изучения динамики популяций в экологии. Однако, для более точного описания динамики популяций необходимо учитывать множество факторов, не входящих в изначальную систему ОДУ, которые могут влиять на популяции в реальных условиях [5].

# 4 Выполнение лабораторной работы

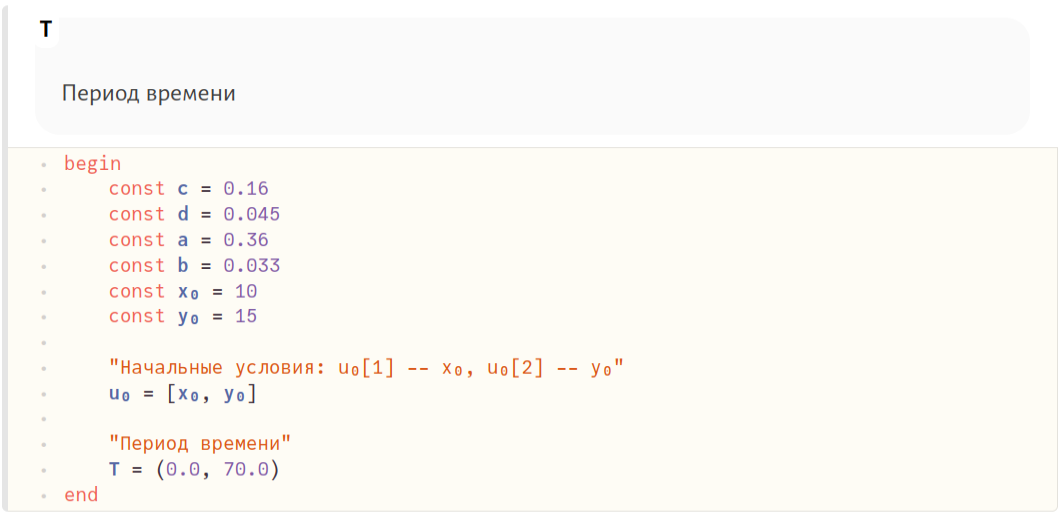
## 4.1 Pluto.jl

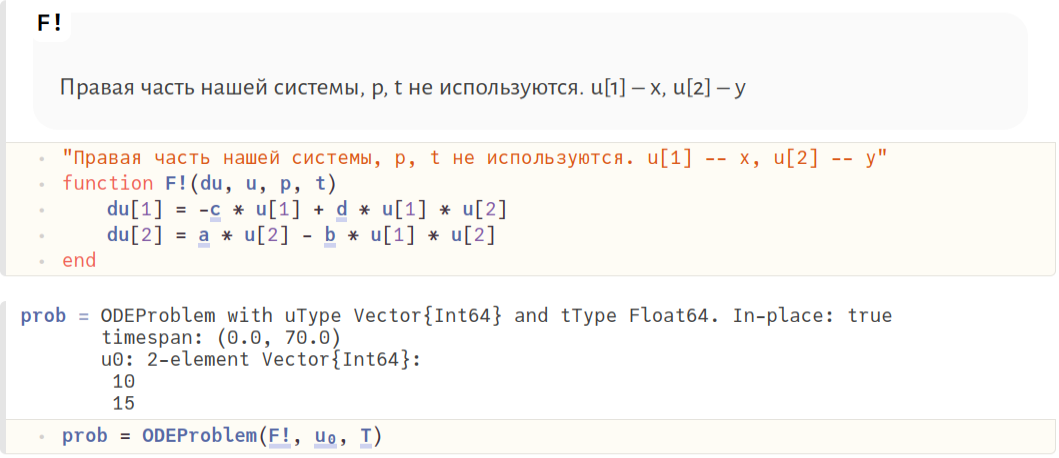
### 4.1.1 Задание №1

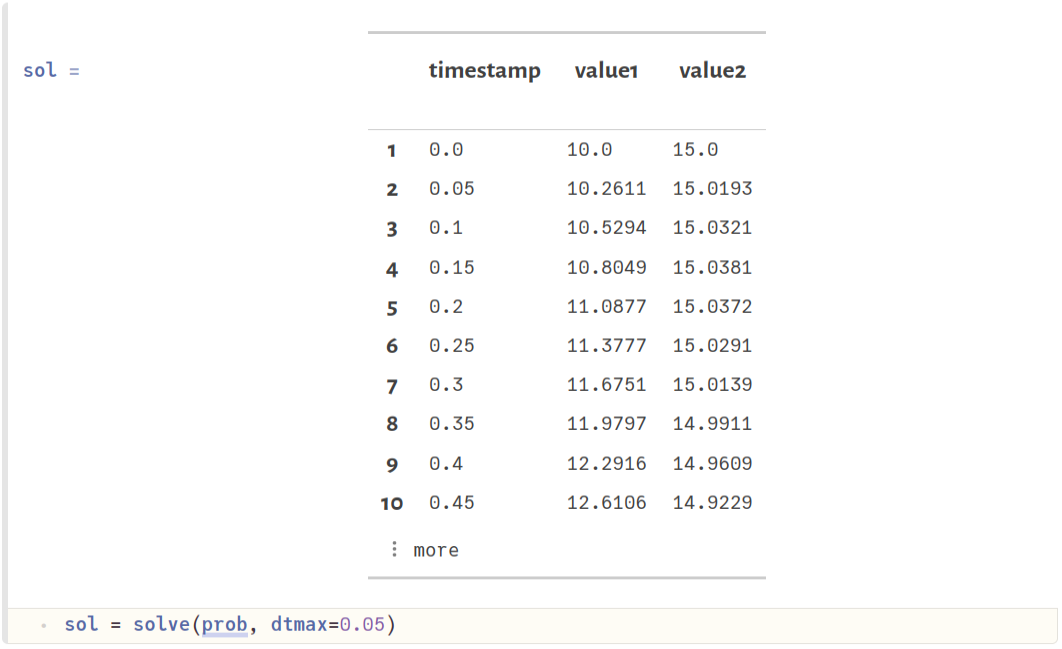
1. Пишем программу, воспроизводящую модель на языке программирования Julia с использованием интерактивного блокнота Pluto (рис. [1](#fig:01), [2](#fig:02), [3](#fig:03), [4](#fig:04), [5](#fig:05), [6](#fig:06)).

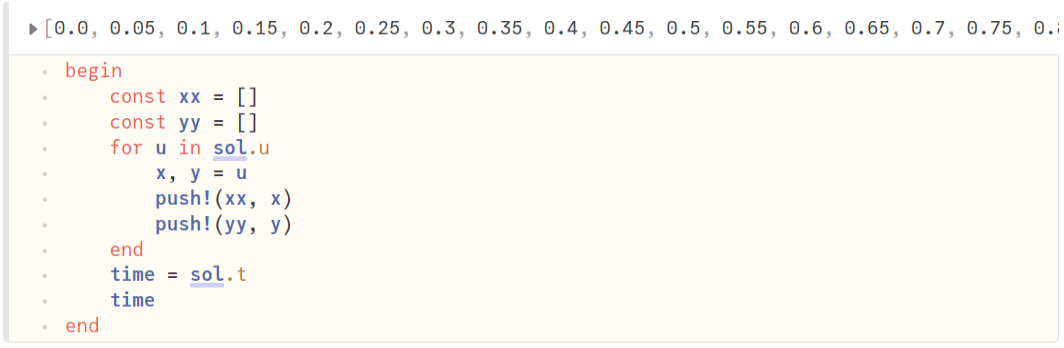
* begin  
   import Pkg  
   Pkg.activate()  
   using DifferentialEquations  
   using LaTeXStrings  
   import Plots  
  end
* begin  
   const c = 0.16  
   const d = 0.045  
   const a = 0.36  
   const b = 0.033  
   const x₀ = 10  
   const y₀ = 15  
    
   "Начальные условия: u₀[1] -- x₀, u₀[2] -- y₀"  
   u₀ = [x₀, y₀]  
    
   "Период времени"  
   T = (0.0, 70.0)  
  end
* "Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"  
  function F!(du, u, p, t)  
   du[1] = -c \* u[1] + d \* u[1] \* u[2]  
   du[2] = a \* u[2] - b \* u[1] \* u[2]  
  end
* prob = ODEProblem(F!, u₀, T)
* sol = solve(prob, dtmax=0.05)
* begin  
   const xx = []  
   const yy = []  
   for u in sol.u  
   x, y = u  
   push!(xx, x)  
   push!(yy, y)  
   end  
   time = sol.t  
   time  
  end
* begin  
   fig = Plots.plot(  
   layout=(1, 2),  
   dpi=150,  
   grid=:xy,  
   gridcolor=:black,  
   gridwidth=1,  
   # aspect\_ratio=:equal,  
   size=(800, 400),  
   legend=:outerbottom,  
   plot\_title="Модель «хищник-жертва»"  
   )  
    
   Plots.plot!(  
   fig[1],  
   time,  
   [xx, yy],  
   color=[:red :blue],  
   xlabel="t",  
   ylabel="x(t), y(t)",  
   label=["x(t) — число хищников" "y(t) — число жертв"]  
   )  
    
   Plots.plot!(  
   fig[2],  
   yy,  
   xx,  
   color=[:grey],  
   xlabel="y(t)",  
   ylabel="x(t)",  
   label="Зависимость числа хищников (x) от числа жертв (y)"  
   )  
  end

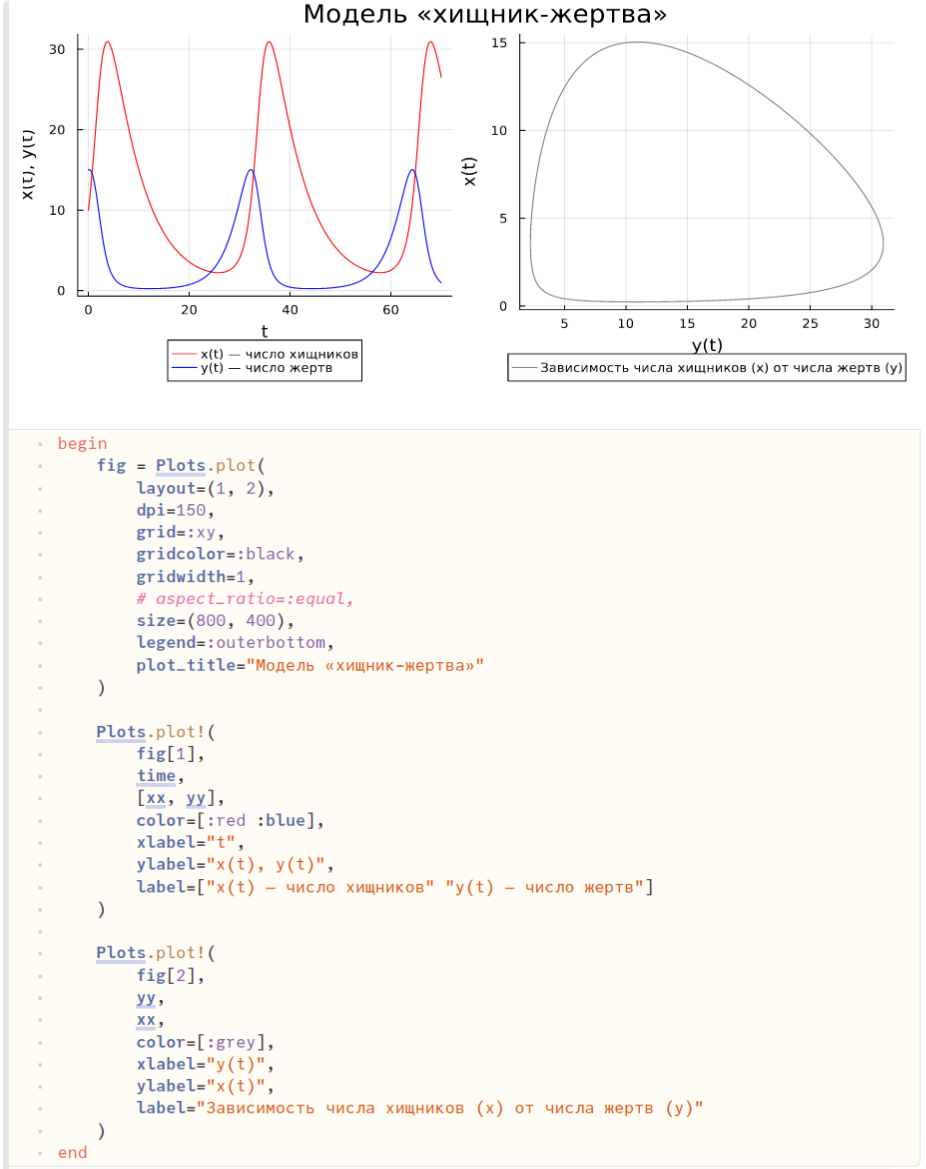
* 
* Figure 1: Импорт библиотек. Задание коэффициентов, начальных условий, периода времени

* 
* Figure 2: Задание коэффициентов, начальных условий, периода времени

* 
* Figure 3: Запись системы уравнений в виде функции. Постановка задачи

* 
* Figure 4: Решение задачи (dtmax отвечает за макс. шаг выбора точек)

* 
* Figure 5: Формирование трех массивов, содержащих значения x, y, t

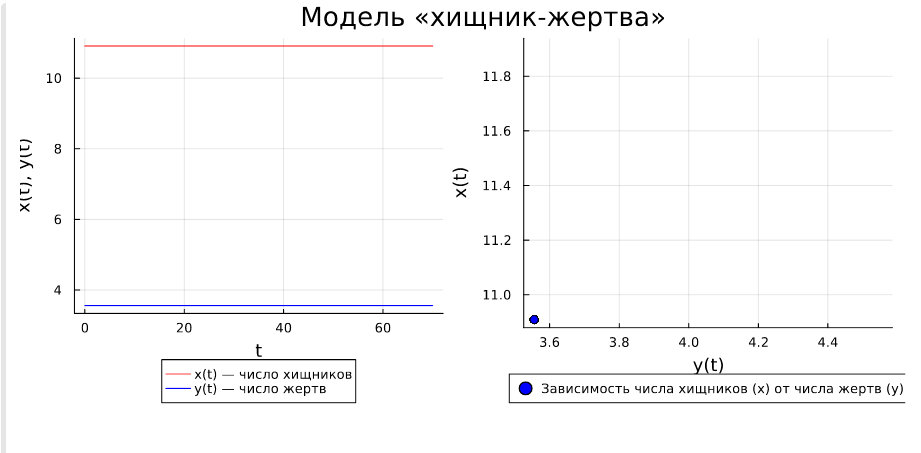
* 
* Figure 6: Отрисовка графиков

### 4.1.2 Задание №2

1. Помимо коэффициентов , , , , определяем начальные условия для стационарной системы и выводим их в консоль. Остальные блоки кода оставляем без изменений. Любуемся результатом (рис. [7](#fig:07), [8](#fig:08)).

* begin  
   const c = 0.16  
   const d = 0.045  
   const a = 0.36  
   const b = 0.033  
   const x₀ = a / b  
   const y₀ = c / d  
    
   @show x₀  
   @show y₀  
    
   "Начальные условия: u₀[1] -- x₀, u₀[2] -- y₀"  
   u₀ = [x₀, y₀]  
    
   "Период времени"  
   T = (0.0, 70.0)  
  end

* 
* Figure 7: Добавление новых начальных условий

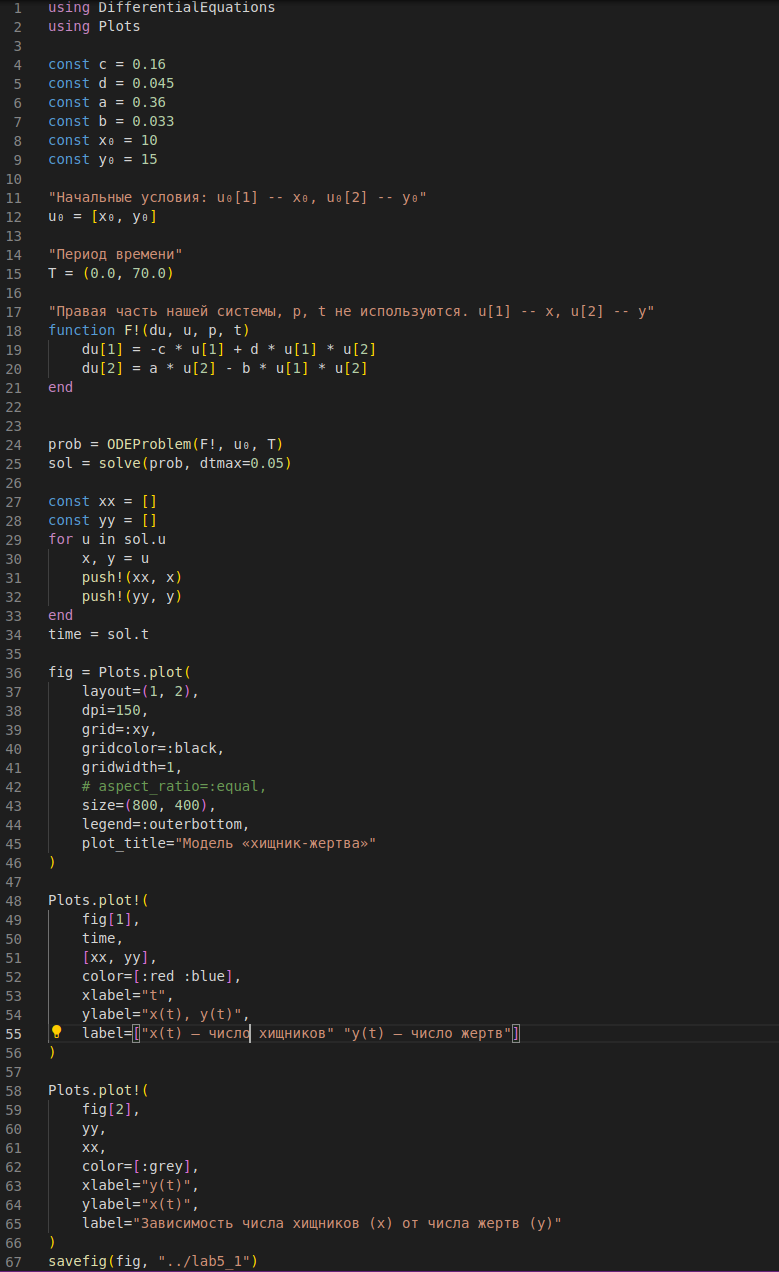
* 
* Figure 8: Результат в виде графиков

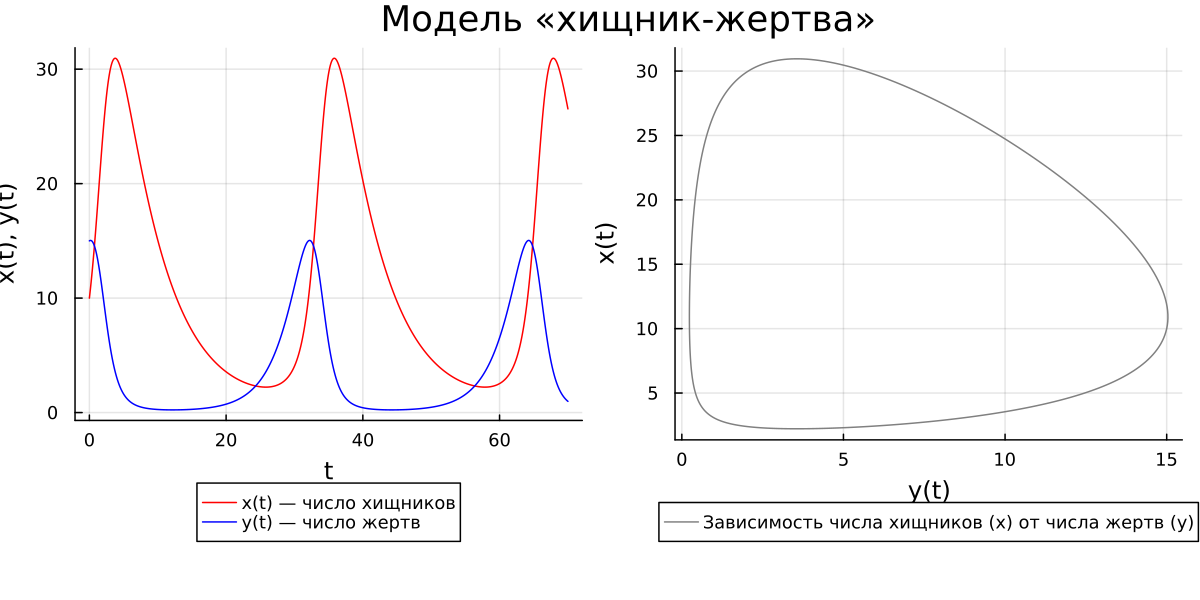
## 4.2 Julia

### 4.2.1 Задание №1

1. Код на Julia в файле аналогичен тому же, написанному с использованием Pluto (рис. [9](#fig:09), [10](#fig:001)). Единственные различия:
   * блоки перенесены в файл в виде построчного алгоритма без повторяющихся ‘begin’ и ‘end’;
   * измененный синтаксис подключения библиотек;
   * выгрузка графиков в виде изображений при помощи метода в последней строчке кода.

* using DifferentialEquations  
  using Plots  
    
  const c = 0.16  
  const d = 0.045  
  const a = 0.36  
  const b = 0.033  
  const x₀ = 10  
  const y₀ = 15  
    
  "Начальные условия: u₀[1] -- x₀, u₀[2] -- y₀"  
  u₀ = [x₀, y₀]  
    
  "Период времени"  
  T = (0.0, 70.0)  
    
  "Правая часть нашей системы, p, t не используются. u[1] -- x, u[2] -- y"  
  function F!(du, u, p, t)  
   du[1] = -c \* u[1] + d \* u[1] \* u[2]  
   du[2] = a \* u[2] - b \* u[1] \* u[2]  
  end  
    
    
  prob = ODEProblem(F!, u₀, T)  
  sol = solve(prob, dtmax=0.05)  
    
  const xx = []  
  const yy = []  
  for u in sol.u  
   x, y = u  
   push!(xx, x)  
   push!(yy, y)  
  end  
  time = sol.t  
    
  fig = Plots.plot(  
   layout=(1, 2),  
   dpi=150,  
   grid=:xy,  
   gridcolor=:black,  
   gridwidth=1,  
   # aspect\_ratio=:equal,  
   size=(800, 400),  
   legend=:outerbottom,  
   plot\_title="Модель «хищник-жертва»"  
  )  
    
  Plots.plot!(  
   fig[1],  
   time,  
   [xx, yy],  
   color=[:red :blue],  
   xlabel="t",  
   ylabel="x(t), y(t)",  
   label=["x(t) — число хищников" "y(t) — число жертв"]  
  )  
    
  Plots.plot!(  
   fig[2],  
   yy,  
   xx,  
   color=[:grey],  
   xlabel="y(t)",  
   ylabel="x(t)",  
   label="Зависимость числа хищников (x) от числа жертв (y)"  
  )  
    
  savefig(fig, "../lab5\_1")

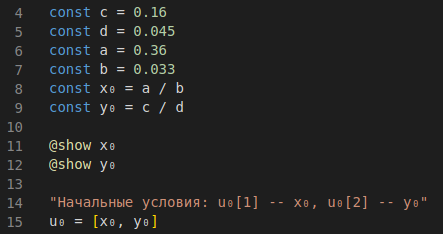
* 
* Figure 9: Код программы на Julia. Аналогичен коду задания для Pluto.jl

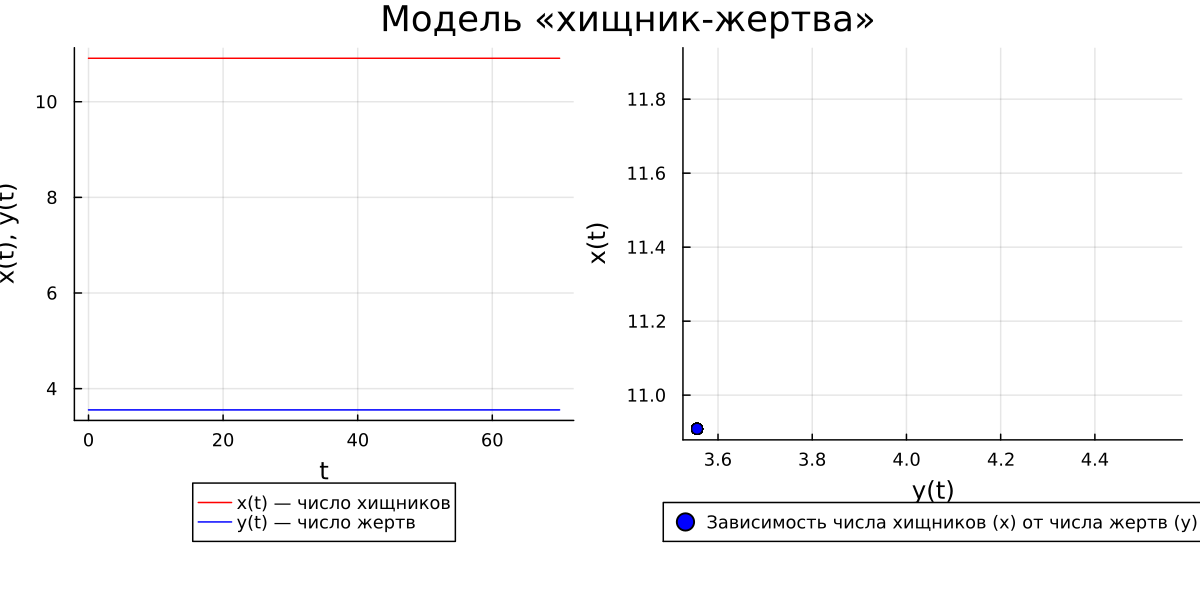
* 
* Figure 10: Результат в виде графиков

### 4.2.2 Задание №2

1. Изменяем необходимые строчки и любуемся результатом (подробное объяснение давалось в предыдущей главе) (рис. [11](#fig:10), [12](#fig:002)).

* const c = 0.16  
  const d = 0.045  
  const a = 0.36  
  const b = 0.033  
  const x₀ = a / b  
  const y₀ = c / d  
    
  @show x₀  
  @show y₀  
    
  "Начальные условия: u₀[1] -- x₀, u₀[2] -- y₀"  
  u₀ = [x₀, y₀]  
    
  "Период времени"  
  T = (0.0, 70.0)

* 
* Figure 11: Измененная часть кода

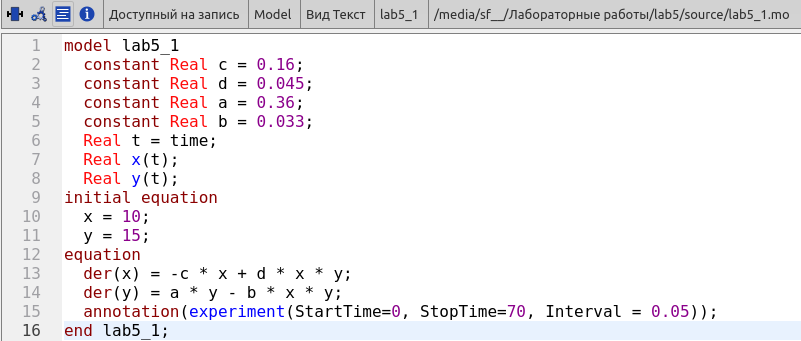
* 
* Figure 12: Результат в виде графиков

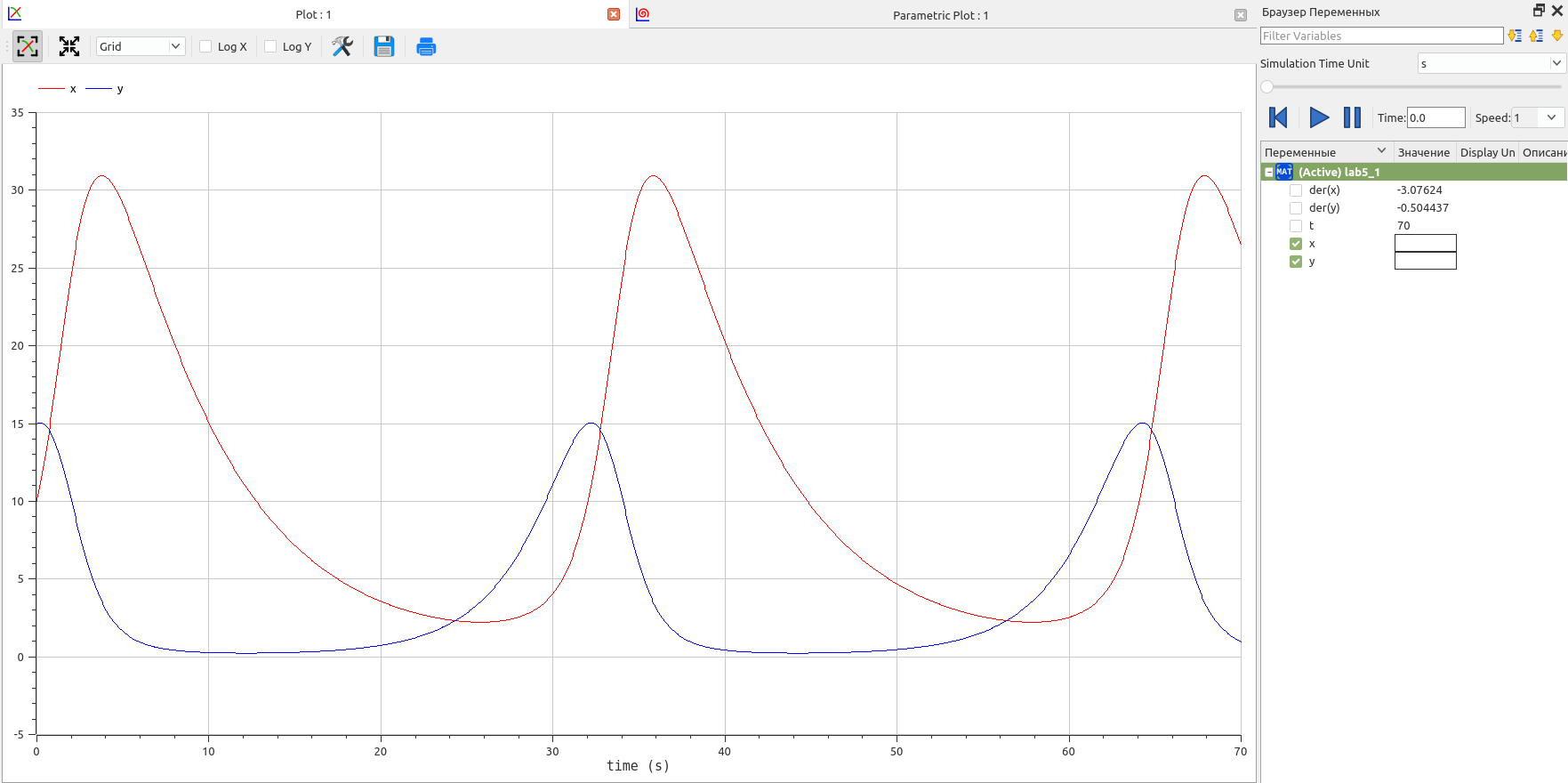
## 4.3 Modelica

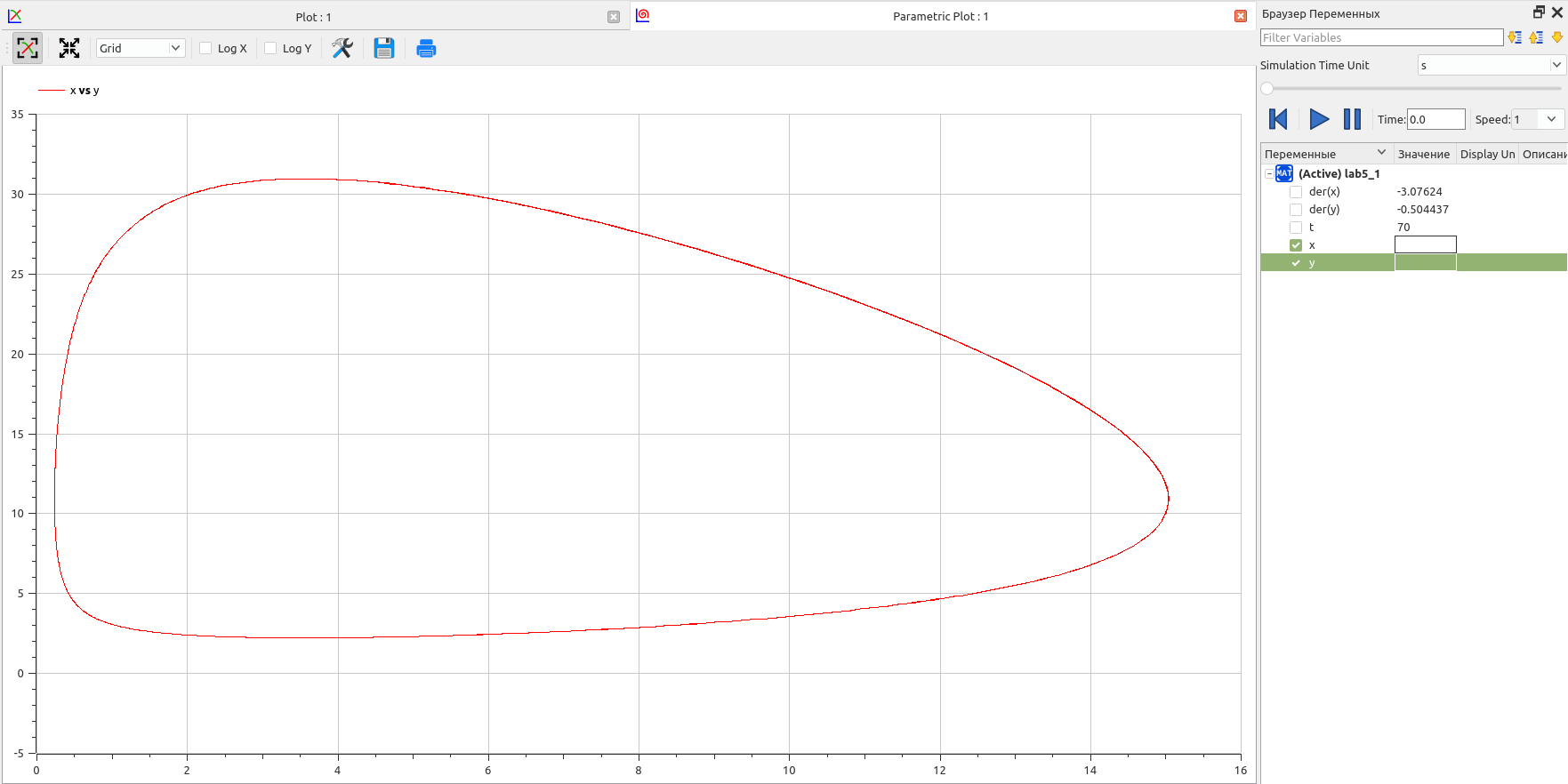
### 4.3.1 Задание №1

1. По аналогии с Julia пишем программу, воспроизводящую модель Лотки-Вольтерры на языке моделирования Modelica с использованием ПО OpenModelica. Любуемся результатами (рис. [13](#fig:11), [14](#fig:12), [15](#fig:13)).

* model lab5\_1  
   constant Real c = 0.16;  
   constant Real d = 0.045;  
   constant Real a = 0.36;  
   constant Real b = 0.033;  
   Real t = time;  
   Real x(t);  
   Real y(t);  
  initial equation  
   x = 10;  
   y = 15;  
  equation  
   der(x) = -c \* x + d \* x \* y;  
   der(y) = a \* y - b \* x \* y;  
   annotation(experiment(StartTime=0, StopTime=70, Interval = 0.05));  
  end lab5\_1;

* 
* Figure 13: Определяем коэффициенты, переменные от времени, начальные условия, систему уравнений, а также начальное/конечное время и частоту разбиения при симуляции

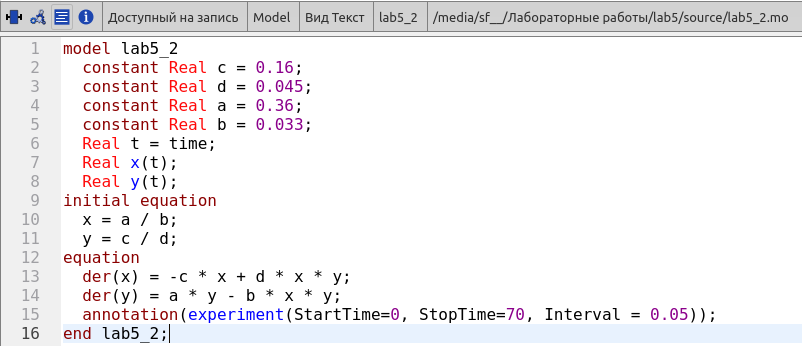
* 
* Figure 14: Результат в виде графика зависимости x и y от t

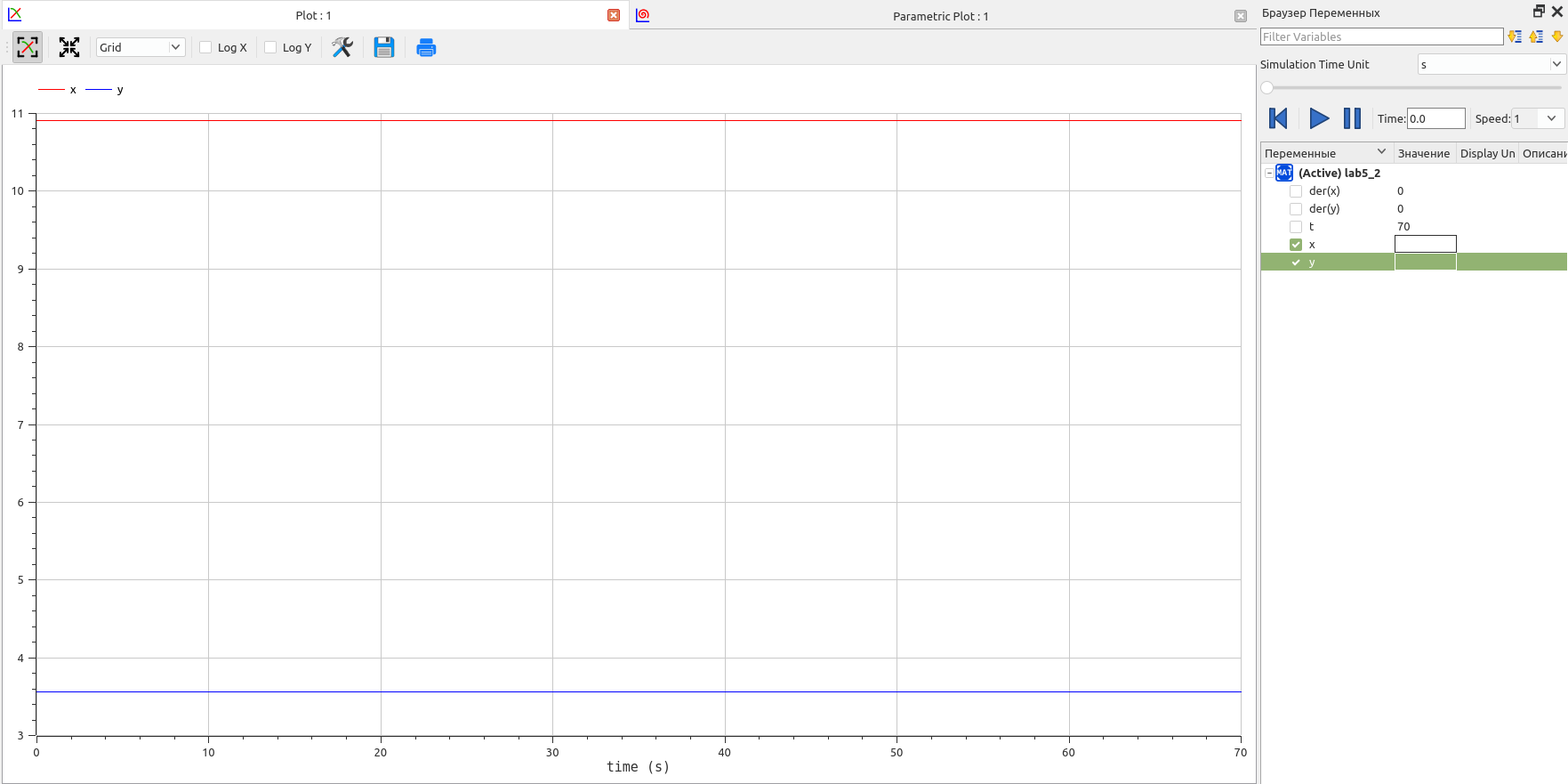
* 
* Figure 15: Результат в виде графика зависимости x (числа хищников) от y (числа жертв)

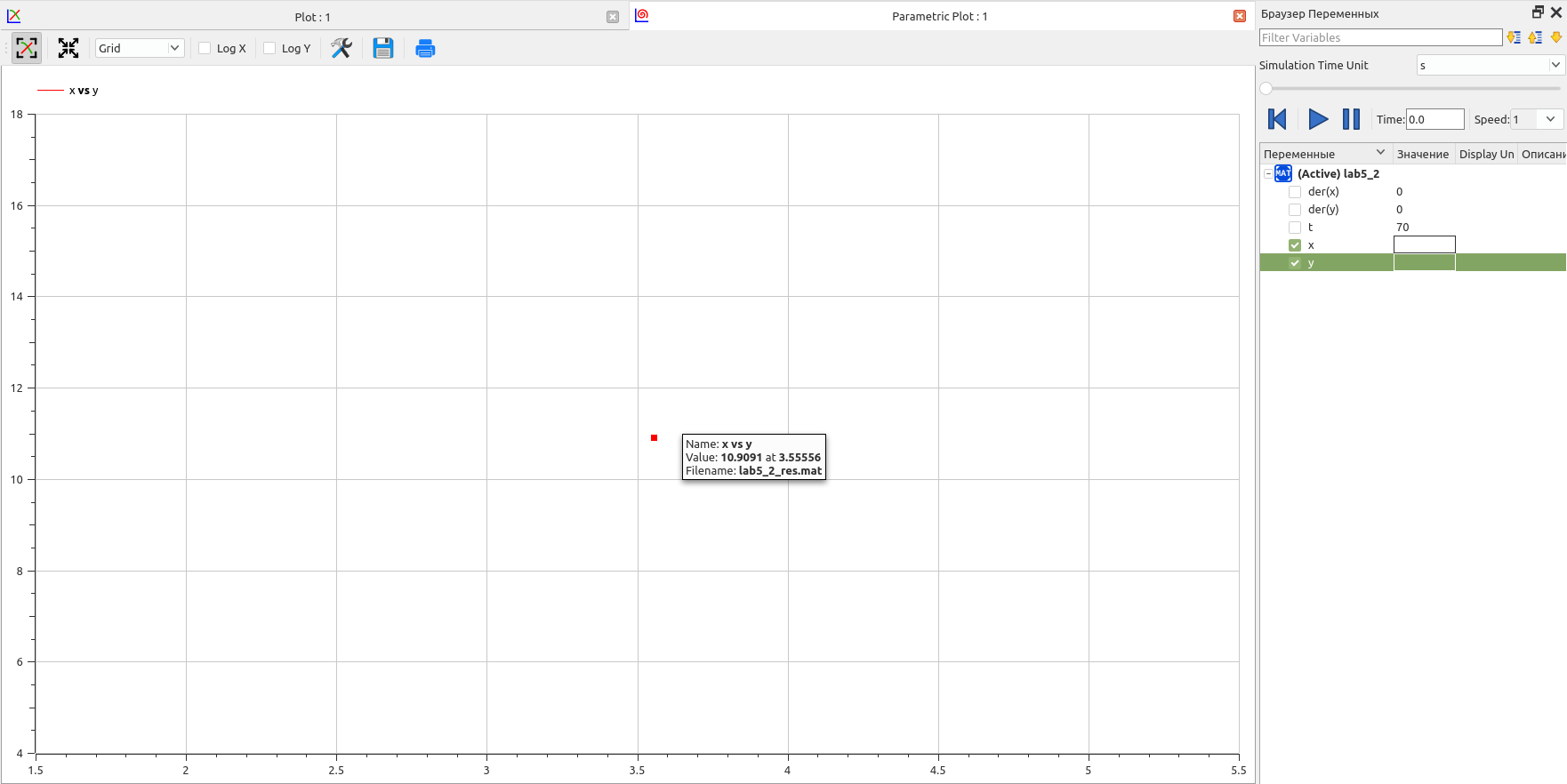
### 4.3.2 Задание №2

1. По аналогии с Julia пишем программу для второго случая. Любуемся результатами (рис. [16](#fig:14), [17](#fig:15), [18](#fig:16)).

* model lab5\_2  
   constant Real c = 0.16;  
   constant Real d = 0.045;  
   constant Real a = 0.36;  
   constant Real b = 0.033;  
   Real t = time;  
   Real x(t);  
   Real y(t);  
  initial equation  
   x = a / b;  
   y = c / d;  
  equation  
   der(x) = -c \* x + d \* x \* y;  
   der(y) = a \* y - b \* x \* y;  
   annotation(experiment(StartTime=0, StopTime=70, Interval = 0.05));  
  end lab5\_2;

* 
* Figure 16: По сравнению с предыдущим случаем изменяются начальные условия на вычисленные

* 
* Figure 17: Результат в виде графика зависимости x и y от t

* 
* Figure 18: Результат в виде графика зависимости y от x

# 5 Анализ результатов

На текущем примере построения математической модели гармонических колебаний мы можем продолжить сравнивать язык программирования Julia и язык моделирования Modelica. Говоря откровенно, по сравнению с анализом результатов при выполнении предыдущей лабораторной работы мало что изменилось: тенденция к сглаживанию негативных моментов при выполнении лабораторной работы на языке программирования Julia продолжается. Со временем и с новыми заданиями, решаемыми при помощи библиотеки DifferentialEquations, скорость написания программ на Julia почти сравнялась с таковой скоростью при использовании Modelica.

Однако, OpenModelica крайне неприятно удивила невозможностью быстрой настройки отрисовки данных на графиках: нет возможности при выполнении второго задания отрисовать жирную точку. Предлагаемое решение OpenModelica (точка, показывающая стационарное состояние системы) почти не видно. Возможно, в будущем я найду решение этой проблемы, однако быстрый поиск по документации пока что результатов не дал.

# 6 Выводы

Продолжил знакомство с функционалом языка программирования Julia, дополнительных библиотек (DifferentialEquations, Plots), интерактивного блокнота Pluto, а также интерактивной командной строкой REPL. Продолжил ознакомление с языком моделирования Modelica и программным обеспечением OpenModelica. Используя эти средства, описал математическую модель Лотки-Вольтерры.

# Список литературы

1. Модель хищник-жертва [Электронный ресурс]. RUDN, 2023. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=967245>.

2. Lotka–Volterra equations [Электронный ресурс]. Wikimedia Foundation, Inc., 2023. URL: <https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations>.

3. Я догоняю, ты убегаешь [Электронный ресурс]. N + 1 Интернет-издание, 2019. URL: <https://nplus1.ru/material/2019/12/04/lotka-volterra-model>.

4. Lotka-Volterra model [Электронный ресурс]. The University of Queensland. URL: <https://teaching.smp.uq.edu.au/scims/Appl_analysis/Lotka_Volterra.html>.

5. Parameter Estimation of the Lotka–Volterra Model [Электронный ресурс]. Ying Hao; Mingshun Guo, 2021. URL: <https://www.feynmanlectures.caltech.edu/I_21.html>.