Отчет по лабораторной работе №5

по дисциплине: Математическое моделирование

Ким Михаил Алексеевич

Содержание

1 Цель работы **4**

2 Задание **5**

3 Теоретическое введение **6**

3.1 Модель «хищник-жертва» (модель Лотки-Вольтерры) 6

3.2 Стационарное состояние системы 7

3.3 Малые изменения параметров в модели «хищник-жертва» 8

4 Выполнение лабораторной работы **10**

4.1 Pluto.jl 10

4.1.1 Задание №1 10

4.1.2 Задание №2 15

4.2 Julia 18

4.2.1 Задание №1 18

4.2.2 Задание №2 22

4.3 Modelica 24

4.3.1 Задание №1 24

4.3.2 Задание №2 26

5 Анализ результатов **29**

6 Выводы **30**

Список литературы **31**

Список иллюстраций

4.1 Импорт библиотек. Задание коэффициентов, начальных условий, периода времени . . . 13
4.2 Задание коэффициентов, начальных условий, периода времени . 13
4.3 Запись системы уравнений в виде функции. Постановка задачи . 13
4.4 Решение задачи (dtmax отвечает за макс. шаг выбора точек) . . . 14
4.5 Формирование трех массивов, содержащих значения x, y, t 14
4.6 Отрисовка графиков . . . 15
4.7 Добавление новых начальных условий . . . 17
4.8 Результат в виде графиков . . . 17
4.9 Код программы на Julia. Аналогичен коду задания для Pluto.jl . . 21
4.10 Результат в виде графиков . . . 22
4.11 Измененная часть кода . . . 23
4.12 Результат в виде графиков . . . 23
4.13 Определяем коэффициенты, переменные от времени, начальные условия, систему уравнений, а также начальное/конечное время и частоту разбиения при симуляции . . . 25
4.14 Результат в виде графика зависимости x и y от t . . . 25
4.15 Результат в виде графика зависимости x (числа хищников) от y (числа жертв) . . . 26
4.16 По сравнению с предыдущим случаем изменяются начальные условия на вычисленные . . . 27
4.17 Результат в виде графика зависимости x и y от t . . . 27
4.18 Результат в виде графика зависимости y от x . . . 28

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»:

$$\begin{cases} \frac{dx}{dt} = -0.16x(t) + 0.045x(t)y(t) \\ \\ \frac{dy}{dt} = 0.36y(t) - 0.033x(t)y(t) \end{cases}$$

Построить график зависимости численности хищников от численности жертв, а также графики изменения численности хищников и численности жертв при следующих начальных условиях: $x_0 = 10, y_0 = 15$. 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 x и хищников y 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:

$$\begin{cases} \frac{dx}{dt} = ax(t) - bx(t)y(t) \\ \\ \frac{dy}{dt} = -cy(t) + dx(t)y(t), \end{cases}$$

где x – число жертв, y - число хищников. Коэффициент a 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 (xy). Каждый акт взаимодействия уменьшает популяцию жертв, но способствует увеличению популяции хищников (члены $-bxy$ и dxy 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 измененным 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. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6).

```
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 x⊠ = 10
    const y⊠ = 15
```

```
    "Начальные условия: u⊠[1] -- x⊠, u⊠[2] -- y⊠"
    u⊠ = [x⊠, y⊠]

    "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!, u⊠, 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
```

```
begin
    import Pkg
    Pkg.activate()
    using DifferentialEquations
    using LaTeXStrings
    import Plots
end
```

```
Activating project at `~/.julia/environments/v1.8`
```

Рис. 4.1: Импорт библиотек. Задание коэффициентов, начальных условий, периода времени

T

Период времени

```
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
```

Рис. 4.2: Задание коэффициентов, начальных условий, периода времени

F!

Правая часть нашей системы, p, t не используются. u[1] – x, u[2] – y

```
"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 with uType Vector{Int64} and tType Float64. In-place: true
       timespan: (0.0, 70.0)
       u0: 2-element Vector{Int64}:
        10
        15
```

```
prob = ODEProblem(F!, u₀, T)
```

Рис. 4.3: Запись системы уравнений в виде функции. Постановка задачи sol =

	timestamp	value1	value2
1	0.0	10.0	15.0
2	0.05	10.2611	15.0193
3	0.1	10.5294	15.0321
4	0.15	10.8049	15.0381
5	0.2	11.0877	15.0372
6	0.25	11.3777	15.0291
7	0.3	11.6751	15.0139
8	0.35	11.9797	14.9911
9	0.4	12.2916	14.9609
10	0.45	12.6106	14.9229

⋮ more

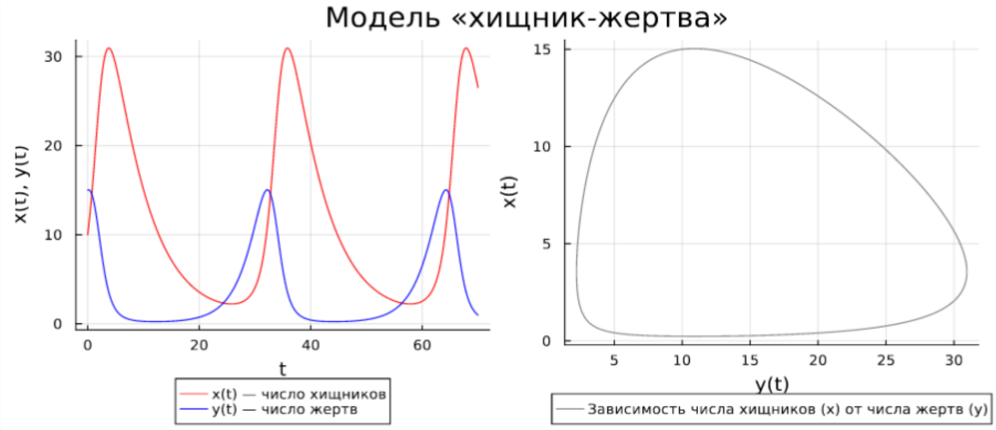
```
sol = solve(prob, dtmax=0.05)
```

Рис. 4.4: Решение задачи (dtmax отвечает за макс. шаг выбора точек)

▸[0.0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.

```
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
```

Рис. 4.5: Формирование трех массивов, содержащих значения x, y, t




```
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
```

Рис. 4.6: Отрисовка графиков

4.1.2 Задание №2

1. Помимо коэффициентов a, b, c, d, определяем начальные условия для стационарной системы и выводим их в консоль. Остальные блоки кода оставляем

без изменений. Любуемся результатом (рис. 4.7, 4.8).

```
begin
    const c = 0.16
    const d = 0.045
    const a = 0.36
    const b = 0.033
    const x⊠ = a / b
    const y⊠ = c / d

    @show x⊠
    @show y⊠

    "Начальные условия: u⊠[1] -- x⊠, u⊠[2] -- y⊠"
    u⊠ = [x⊠, y⊠]

    "Период времени"
    T = (0.0, 70.0)
end
```

T

Период времени

```
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
```

```
x₀ = 10.909090909090908
y₀ = 3.5555555555555556
```

Рис. 4.7: Добавление новых начальных условий

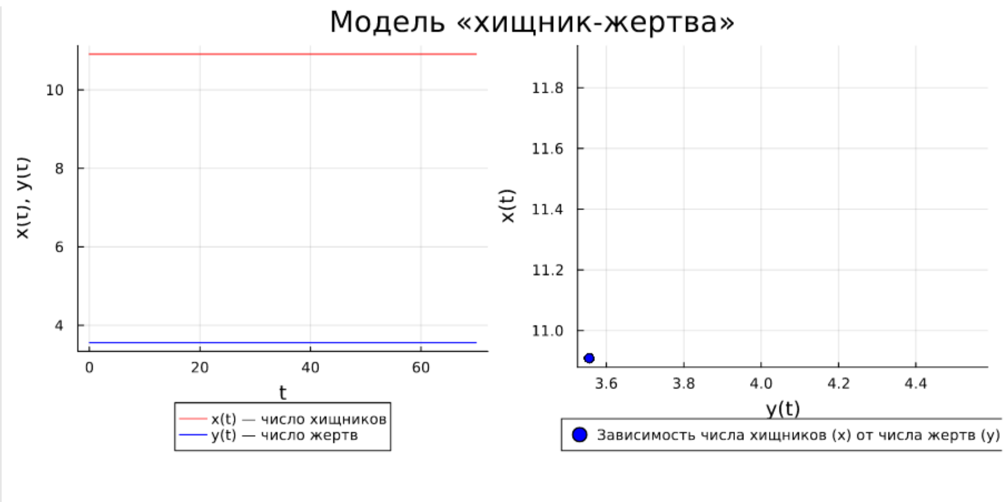


Рис. 4.8: Результат в виде графиков

4.2 Julia

4.2.1 Задание №1

1. Код на Julia в файле аналогичен тому же, написанному с использованием Pluto (рис. 4.9, 4.10). 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 x⊠ = 10
const y⊠ = 15

"Начальные условия: u⊠[1] -- x⊠, u⊠[2] -- y⊠"
u⊠ = [x⊠, y⊠]

"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!, u⊠, 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")
```

```julia
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")
```

Рис. 4.9: Код программы на Julia. Аналогичен коду задания для Pluto.jl

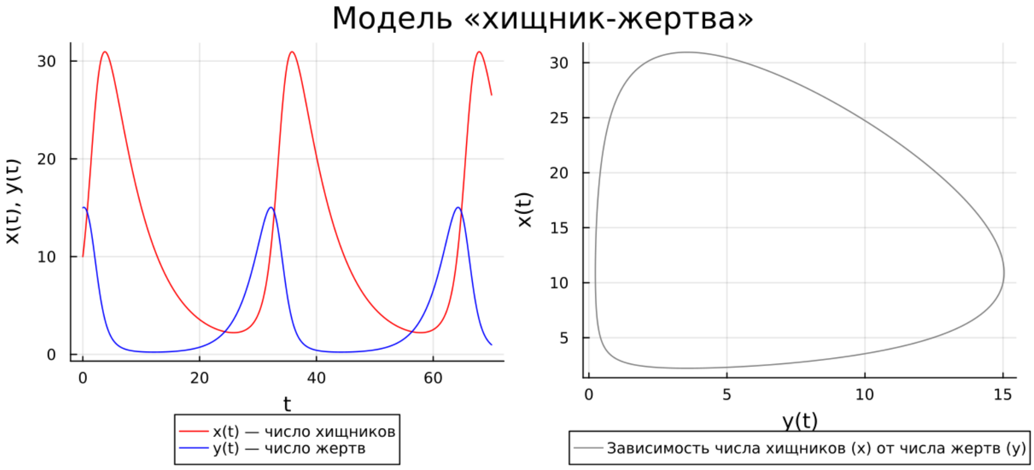


Рис. 4.10: Результат в виде графиков

4.2.2 Задание №2

1. Изменяем необходимые строчки и любуемся результатом (подробное объяснение давалось в предыдущей главе) (рис. 4.11, 4.12).

```
const c = 0.16
const d = 0.045
const a = 0.36
const b = 0.033
const x⊠ = a / b
const y⊠ = c / d

@show x⊠
@show y⊠

"Начальные условия: u⊠[1] -- x⊠, u⊠[2] -- y⊠"
u⊠ = [x⊠, y⊠]
```

```
"Период времени"
T = (0.0, 70.0)
```

```
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]
```

Рис. 4.11: Измененная часть кода

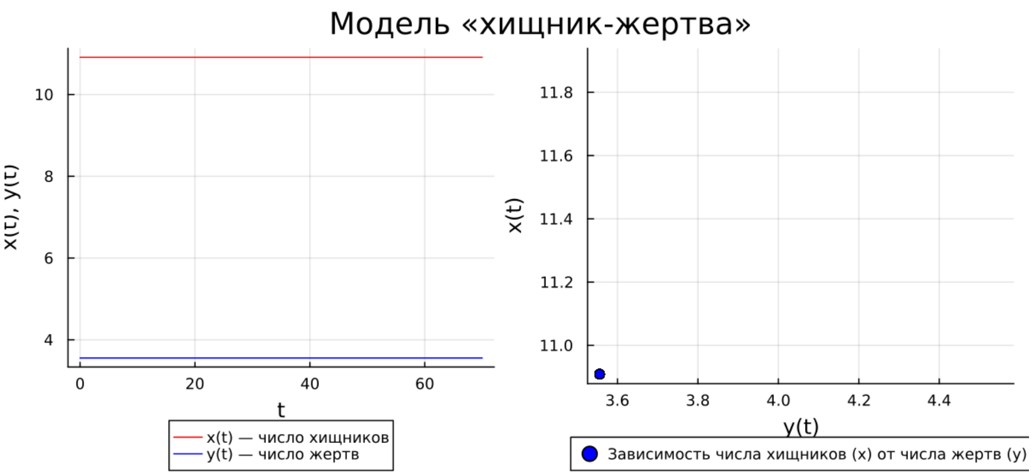


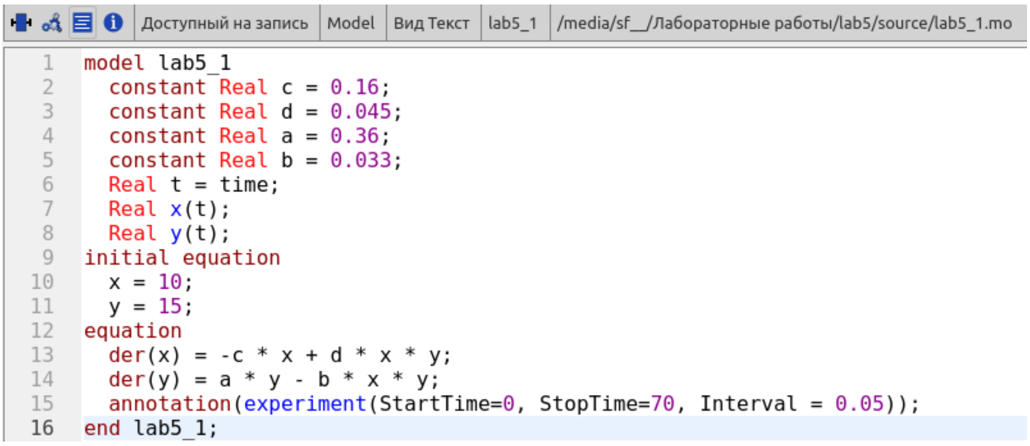
Рис. 4.12: Результат в виде графиков

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. 4.13, 4.14, 4.15).

```
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;
```




```
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;
```

Рис. 4.13: Определяем коэффициенты, переменные от времени, начальные условия, систему уравнений, а также начальное/конечное время и частоту разбиения при симуляции

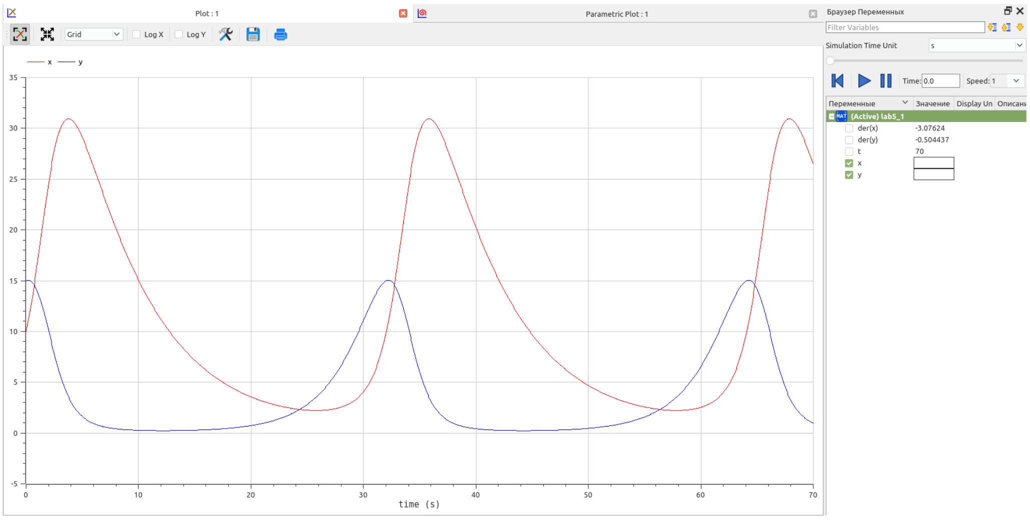


Рис. 4.14: Результат в виде графика зависимости x и y от t

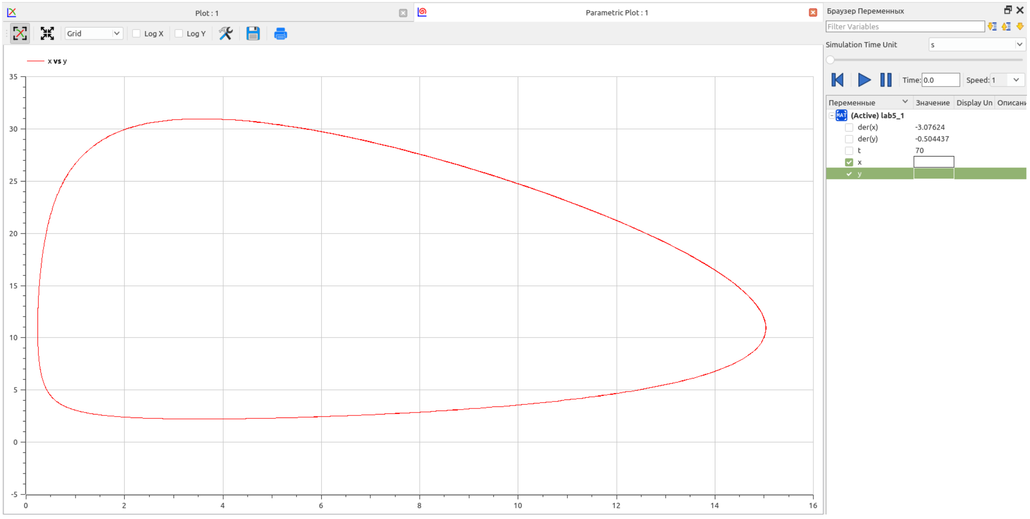


Рис. 4.15: Результат в виде графика зависимости x (числа хищников) от y (числа жертв)

4.3.2 Задание №2

1. По аналогии с Julia пишем программу для второго случая. Любуемся результатами (рис. 4.16, 4.17, 4.18).

```
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;
```

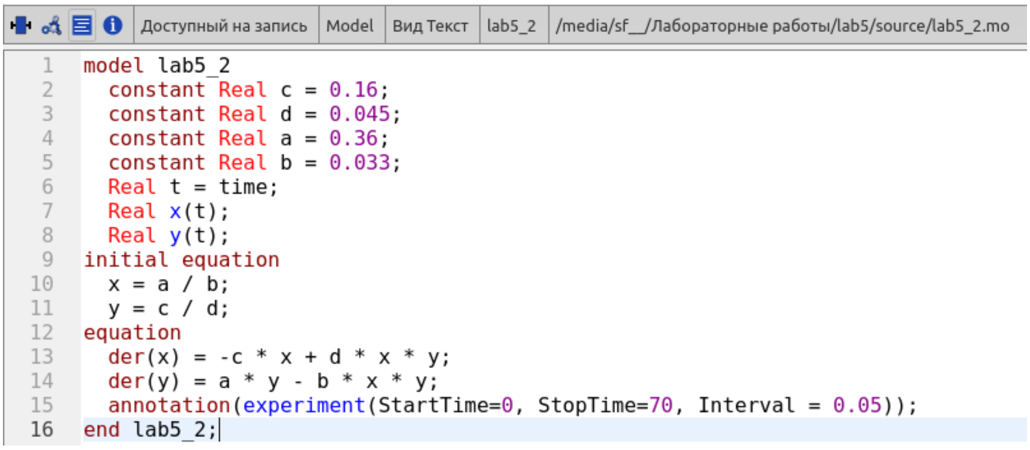


Рис. 4.16: По сравнению с предыдущим случаем изменяются начальные условия на вычисленные

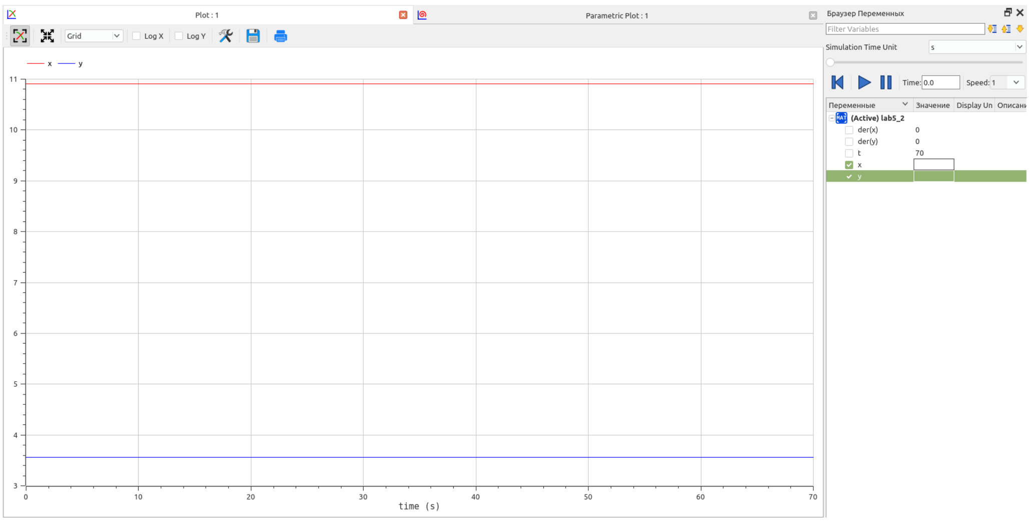


Рис. 4.17: Результат в виде графика зависимости x и y от t

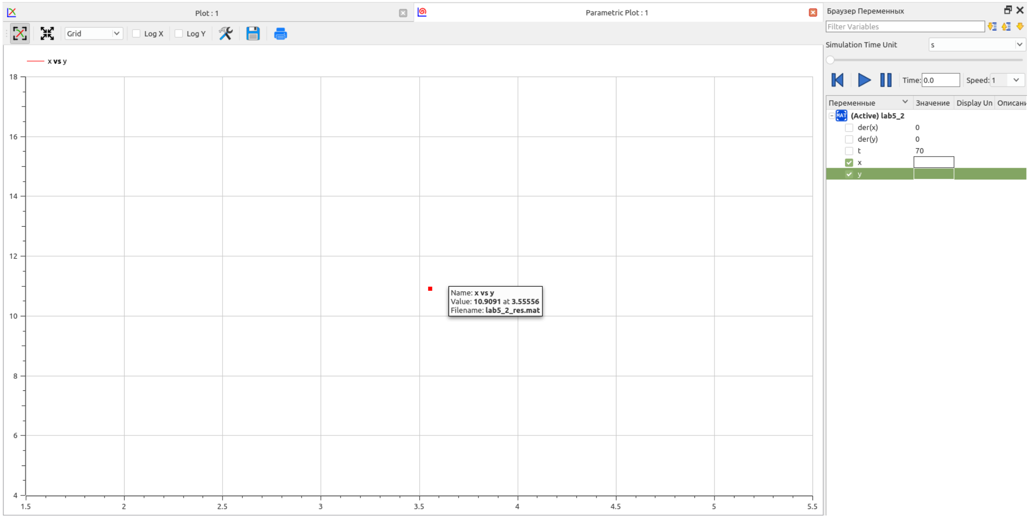


Рис. 4.18: Результат в виде графика зависимости y от x

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.