Отчёт по лабораторной работе 5

Модель хищник-жертва

Аристова Арина Олеговна

Содержание

# 1 Цель работы

Цель:

* Изучить простейшую модель взаимодействия двух видов типа «хищник — жертва» - модель Лотки-Вольтерры.
* Построить график зависимости x от y и графики функций
* Найти стационарное состояние системы

# 2 Задание

## 2.1 Вариант 4

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

Julia – высокоуровневый язык, который разработан для научного программирования. Язык поддерживает широкий функционал для математических вычислений и работы с большими массивами данных[1].

OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. Активно развивается Open Source Modelica Consortium, некоммерческой неправительственной организацией. Open Source Modelica Consortium является совместным проектом RISE SICS East AB и Линчёпингского университета. По своим возможностям приближается к таким вычислительным средам как Matlab Simulink, Scilab xCos, имея при этом значительно более удобное представление системы уравнений исследуемого блока [2].

* Модель Лотки—Вольтерры — модель взаимодействия двух видов типа «хищник — жертва», названная в честь её авторов, которые предложили модельные уравнения независимо друг от друга. Такие уравнения можно использовать для моделирования систем «хищник — жертва», «паразит — хозяин», конкуренции и других видов взаимодействия между двумя видами. [3]

Простейшая модель взаимодействия двух видов типа «хищник — жертва» - модель Лотки-Вольтерры. Данная двувидовая 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

В этой модели x – число жертв, y - число хищников. Коэффициент a описывает скорость естественного прироста числа жертв в отсутствие хищников, с - 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 и в правой части уравнения).

Математический анализ этой (жесткой) модели показывает, что имеется стационарное состояние (A на рис. 3.1), всякое же другое начальное состояние (B) приводит к периодическому колебанию численности как жертв, так и хищников, так что по прошествии некоторого времени система возвращается в состояние B.

Стационарное состояние системы (положение равновесия, не зависящее от времени решение) будет в точке: . Если начальные значения задать в стационарном состоянии x , то в любой момент времени численность популяций изменяться не будет. При малом отклонении от положения равновесия численности как хищника, так и жертвы с течением времени не возвращаются к равновесным значениям, а совершают периодические колебания вокруг стационарной точки. Амплитуда колебаний и их период определяется начальными значениями численностей . Колебания совершаются в противофазе[3,4].

При малом изменении модели

(прибавление к правым частям малые члены, учитывающие, например, конкуренцию жертв за пищу и хищников за жертв), вывод о периодичности (возвращении системы в исходное состояние B), справедливый для жесткой системы Лотки-Вольтерры, теряет силу. Таким образом, мы получаем так называемую мягкую модель «хищник-жертва».

# 4 Выполнение лабораторной работы

## 4.1 Выполнение на Julia

Мой вариант лабораторной работы: 4. Я получила его по заданной формуле:

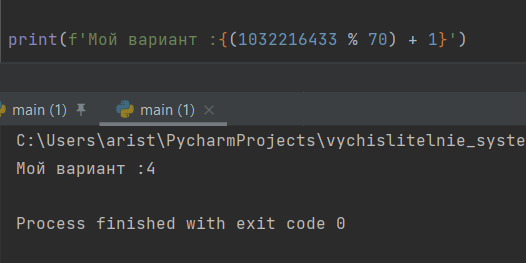


Рис. 1: Определение варианта.

Затем я написала 2 программы для каждого из случаев на языке Julia:

Вот листинг первой программы для **нестационарного** случая. Проблема заключается аналогично предыдущим лабораторным работам в решении одногодного дифференциального уравнения. Решение этой проблемы и отображается на графике.

using Plots  
using DifferentialEquations  
  
x0 = 9  
y0 = 14  
  
a = 0.15  
b = 0.044  
c = 0.35  
d = 0.032  
  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a\*u[1] + b \* u[1] \* u[2]  
 du[2] = c \* u[2] - d \* u[1] \* u[2]  
end  
  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
prob = ODEProblem(ode\_fn, v0, tspan)  
sol = solve(prob, dtmax=0.05)  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
T = [t for t in sol.t]  
  
plt1 = plot(dpi=300, legend=false)  
  
plot!(plt1, title="График зависимости x от y")  
plot!(plt1, X, Y, color=:green)  
  
savefig(plt1, "lab5\_1\_1.png")  
  
plt2 = plot(dpi=300, legend=true)  
  
plot!(plt2, title="Нестационарный случай")  
plot!(plt2, T, X, label="Численность жертв", color=:blue)  
  
plot!( plt2, T, Y, label="Численность хищников", color=:red)  
  
savefig(plt2, "lab5\_1\_2.png")

Полученный результат:

* Первый график отражает :

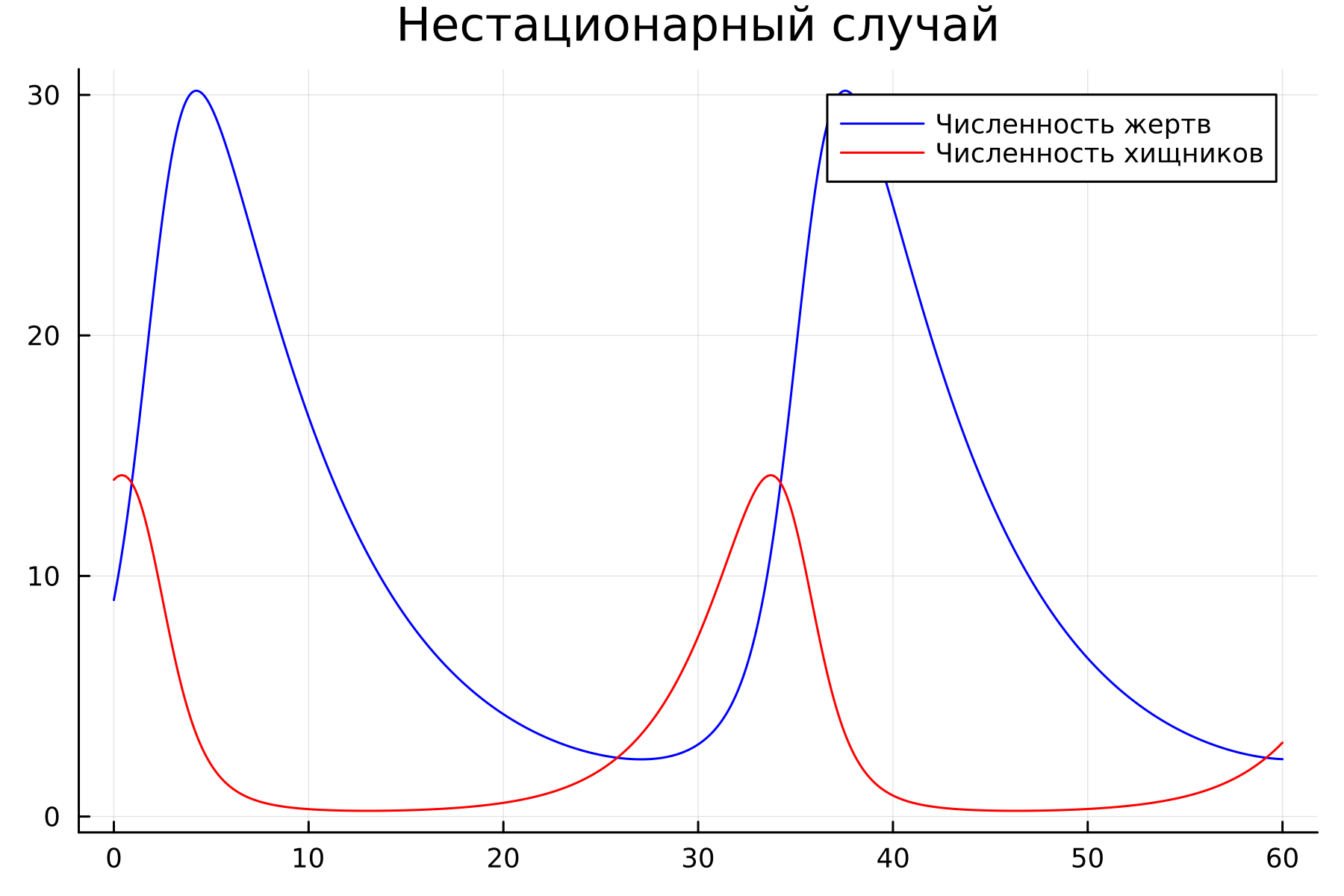


Рис. 2: Случай 1.Решение, полученное на Julia

* Второй график отражает зависимость x от y:



Рис. 3: Случай 1. График зависимости x от y, полученный на Julia

Вот листинг второй программы, который помог нам отыскать **стационароное состояние системы**: Стационарное состояние системы (положение равновесия, не зависящее от времени решение) достигается в точке: .

using Plots  
using DifferentialEquations  
  
a = 0.15  
b = 0.044  
c = 0.35  
d = 0.032  
  
x0 = c / d  
y0 = a / b  
  
function ode\_fn(du, u, p, t)  
 x, y = u  
 du[1] = -a\*u[1] + b \* u[1] \* u[2]  
 du[2] = c \* u[2] - d \* u[1] \* u[2]  
end  
  
v0 = [x0, y0]  
tspan = (0.0, 60.0)  
prob = ODEProblem(ode\_fn, v0, tspan)  
sol = solve(prob, dtmax=0.05)  
X = [u[1] for u in sol.u]  
Y = [u[2] for u in sol.u]  
T = [t for t in sol.t]  
  
plt = plot(dpi=300, legend=true)  
  
plot!(plt, title="Cтационарный случай")  
plot!(plt, T, X, label="Численность жертв", color=:blue)  
  
plot!( plt, T, Y, label="Численность хищников", color=:red)  
  
savefig(plt, "lab5\_2.png")

Полученный результат:

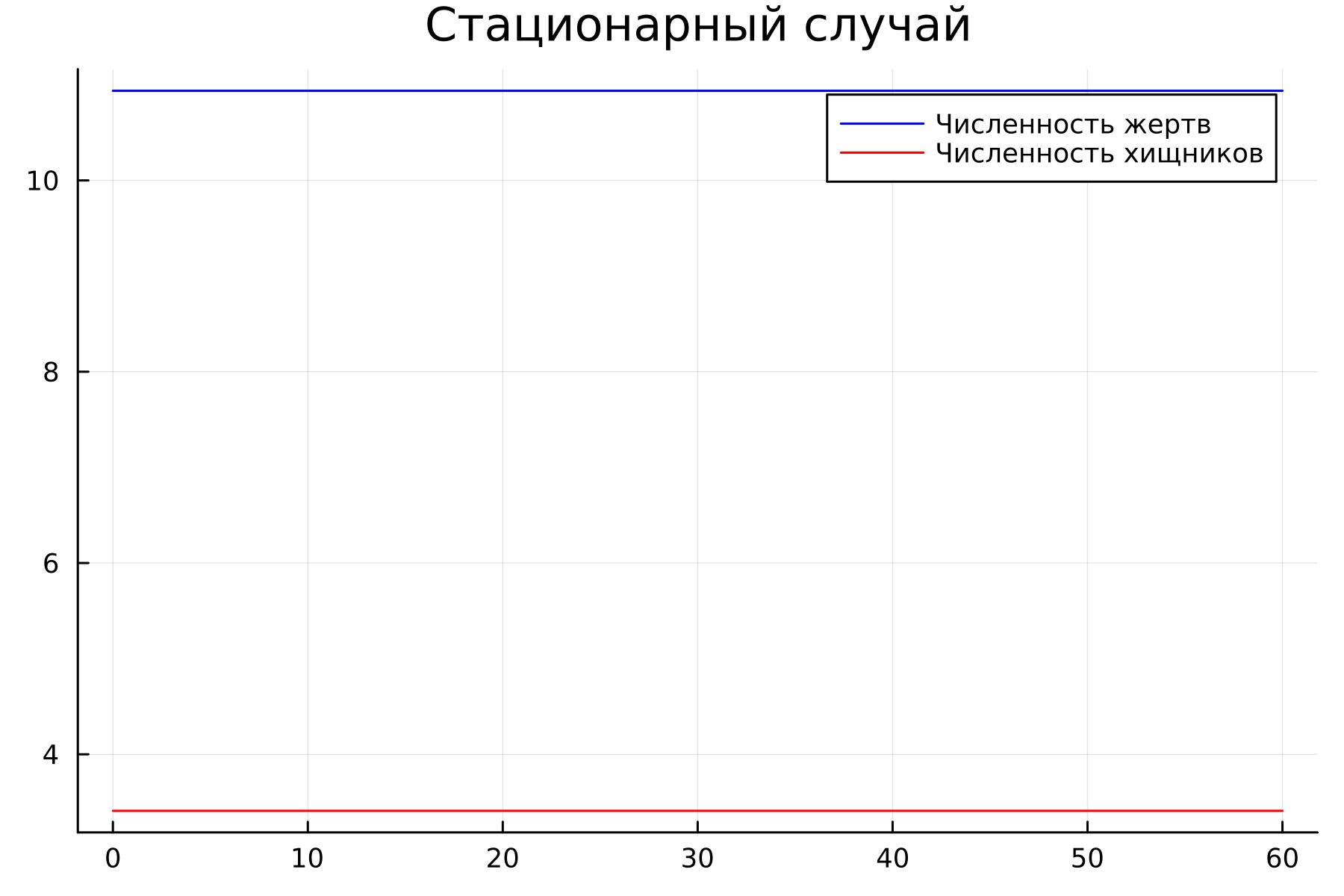


Рис. 4: Случай 2. Стационарное состояние, полученное на Julia

## 4.2 Выполнение на Modelica

Затем я написала необходимые программы для каждого из случаев для получения решений на языке Modelica в OpenModelica:

Вот листинг первой программы для **нестационарного случая** случая:

model lab5\_1  
  
Real a = 0.15;  
Real b = 0.044;  
Real c = 0.35;  
Real d = 0.032;  
  
Real x;  
Real y;  
  
initial equation  
x = 9;  
y = 14;  
  
equation  
der(x) = -a\*x + b\*x\*y;  
der(y) = c\*y - d\*x\*y;  
end lab5\_1;

Полученный результат:

* Первый график отражает зависимость :

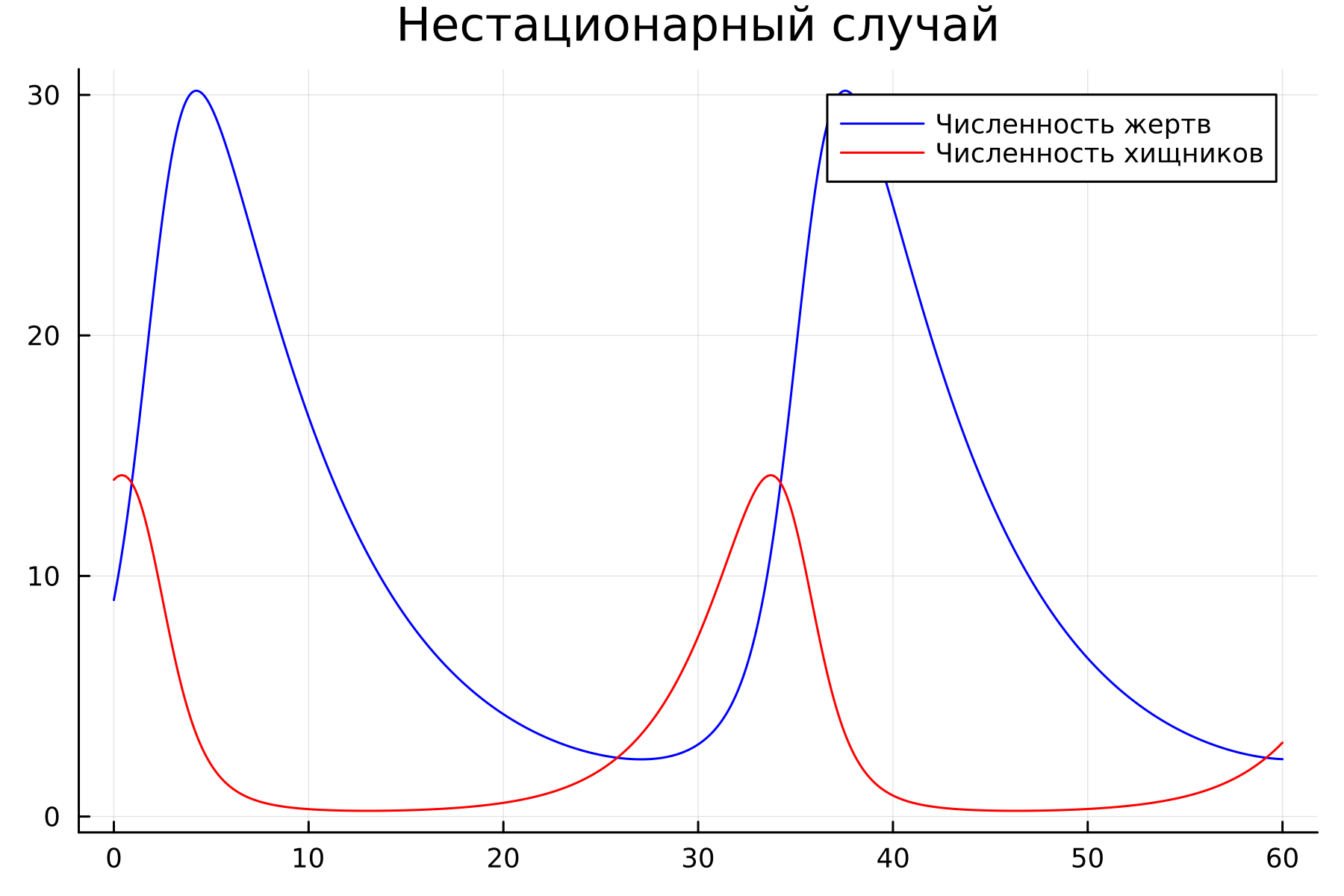


Рис. 5: Случай 1. Решение, полученное на Modelica

* Второй график отражает зависимость x от y:



Рис. 6: Случай 1. График зависимости x от y, полученный на Modelica

Вот листинг второй программы, который помог нам отыскать **стационароное состояние системы**: Стационарное состояние системы (положение равновесия, не зависящее от времени решение) достигается в точке: .

model lab5\_2  
  
Real a = 0.15;  
Real b = 0.044;  
Real c = 0.35;  
Real d = 0.032;  
  
Real x;  
Real y;  
  
initial equation  
x = c / d;  
y = a / b;  
  
equation  
der(x) = -a\*x + b\*x\*y;  
der(y) = c\*y - d\*x\*y;  
end lab5\_2;

Полученный результат:

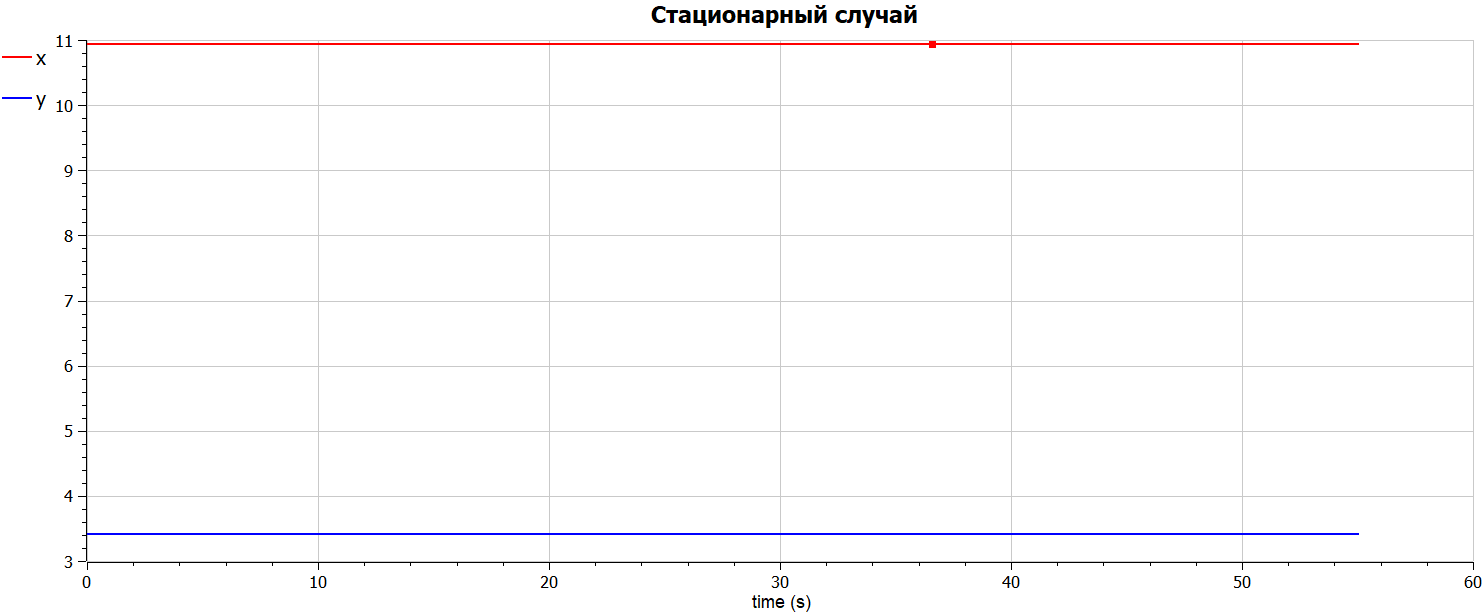


Рис. 7: Случай 2. Стационарное состояние, полученное на Julia

# 5 Анализ полученных результатов

В итоге проделанной работы мною были построены все необходимые графики: график зависимости численности хищников от численности жертв, графики изменения численности хищников и численности жертв на языках Julia и OpenModelica. Аналогично предыдущим лабораторным работам, код для построения модели хищник-жертва на языке Modelica занимает меньше строк, нежели аналогичное построение на Julia.

Также немаловажно, что на обоих языках содержание графиков получилось идентичным:

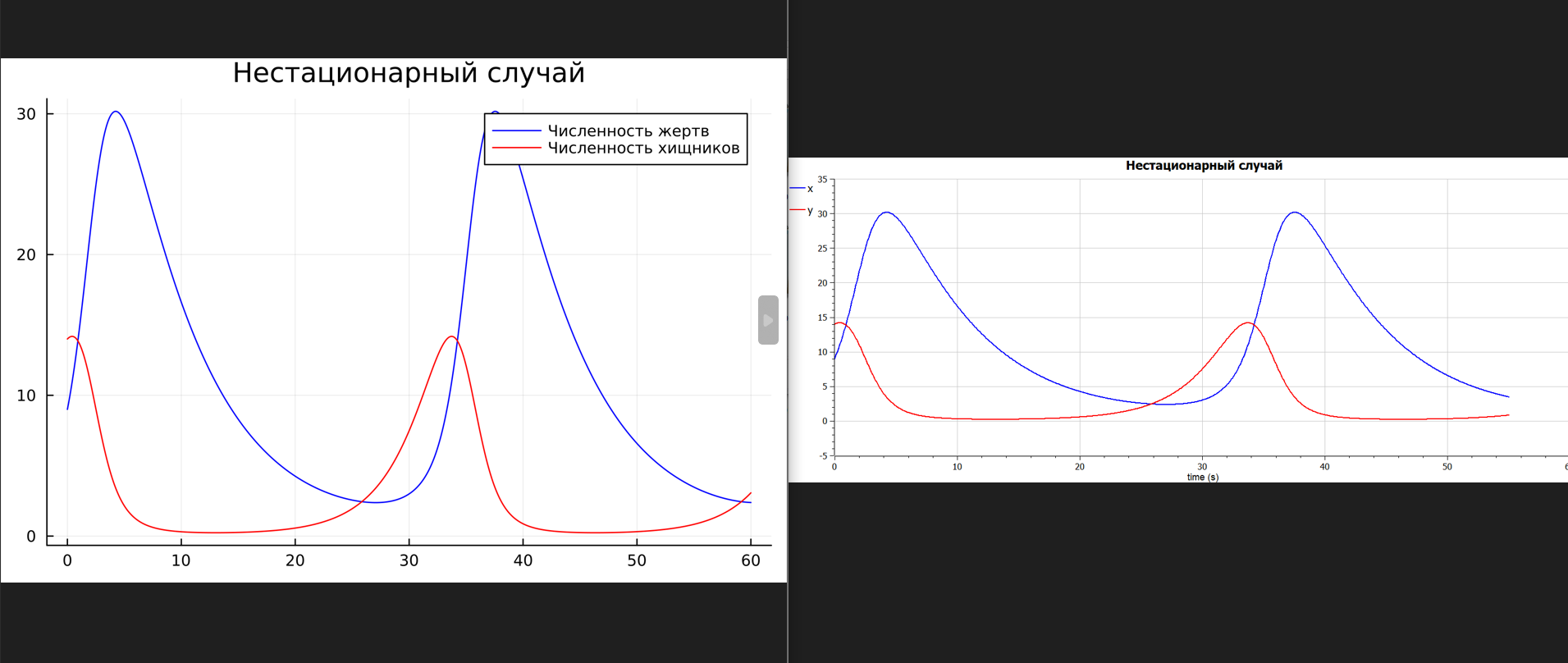


Рис. 8: Сравнение графиков нестационарного состояния.

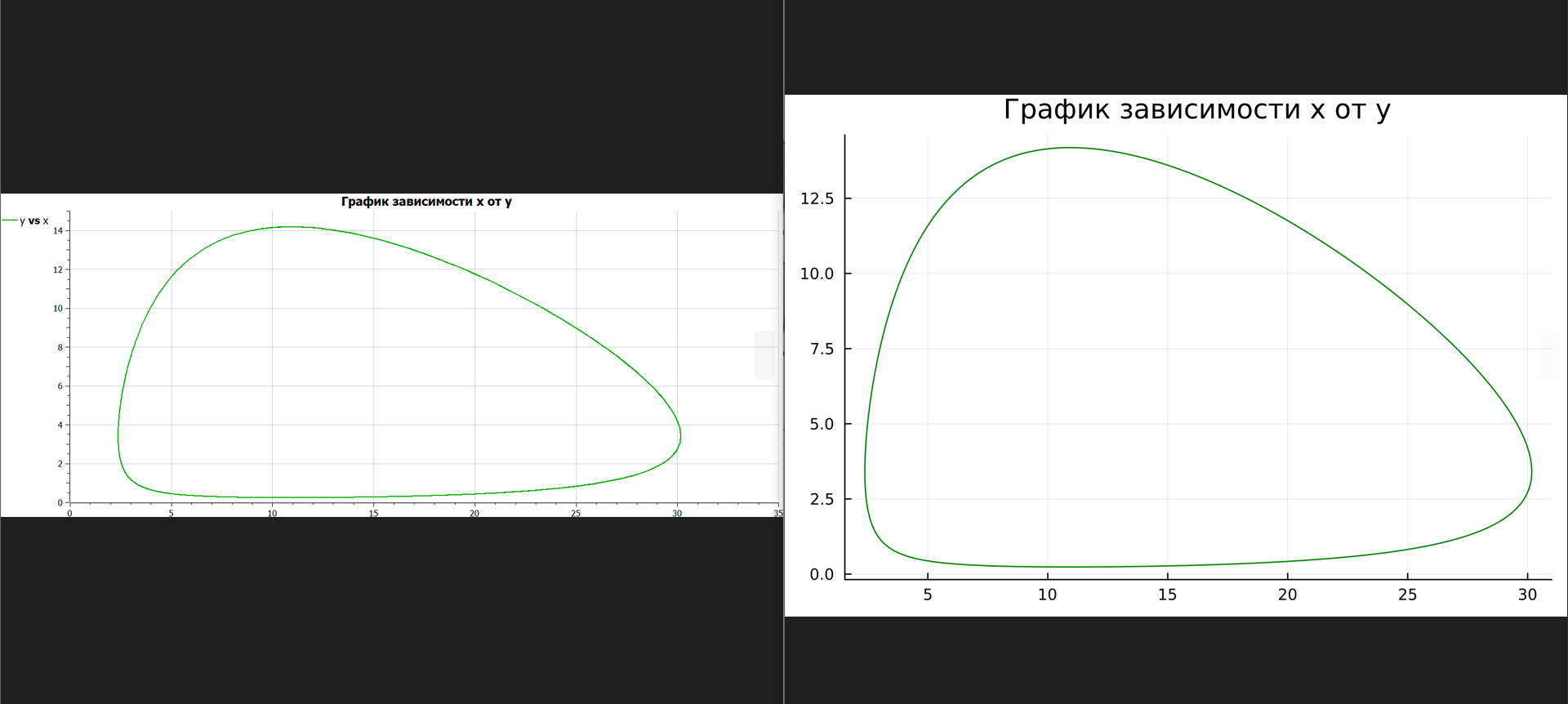


Рис. 9: Сравнение графиков зависимости x от y.

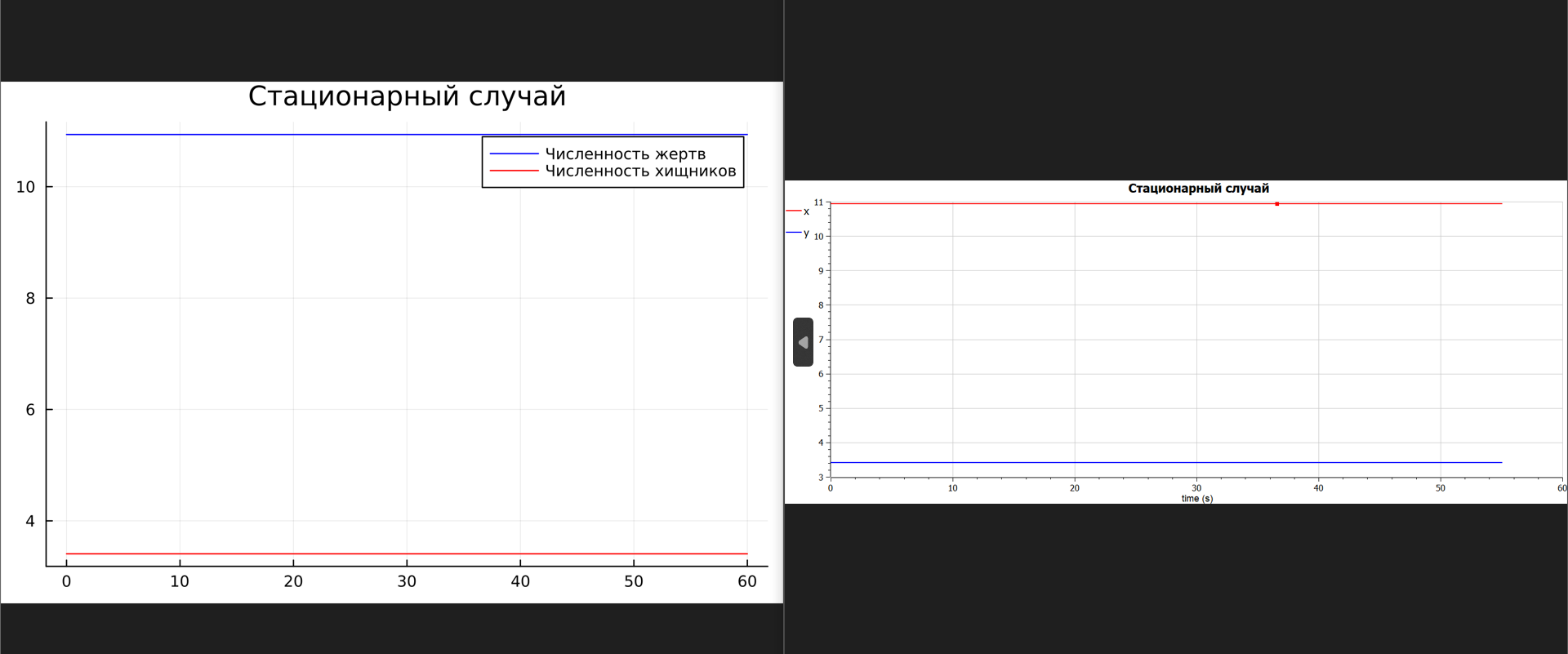


Рис. 10: Сравнение графиков стационарного состояния.

# 6 Выводы

В ходе и по результатам выполнения лабораторной работы я построила необходимые графики (два описывающие нестационарный случай и зависимость количества жертв от количества хищников, и описывающий стационарный случай) на двух языках: Julia и Modelica.

# Список литературы. Библиография

[1] Документация по Julia: https://docs.julialang.org/en/v1/

[2] Документация по OpenModelica: https://openmodelica.org/

[3] Модель Лотки – Вольтерры https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C\_%D0%9B%D0%BE%D1%82%D0%BA%D0%B8\_%E2%80%94\_%D0%92%D0%BE%D0%BB%D1%8C%D1%82%D0%B5%D1%80%D1%80%D1%8B

[4] Материалы к лабораторной работе