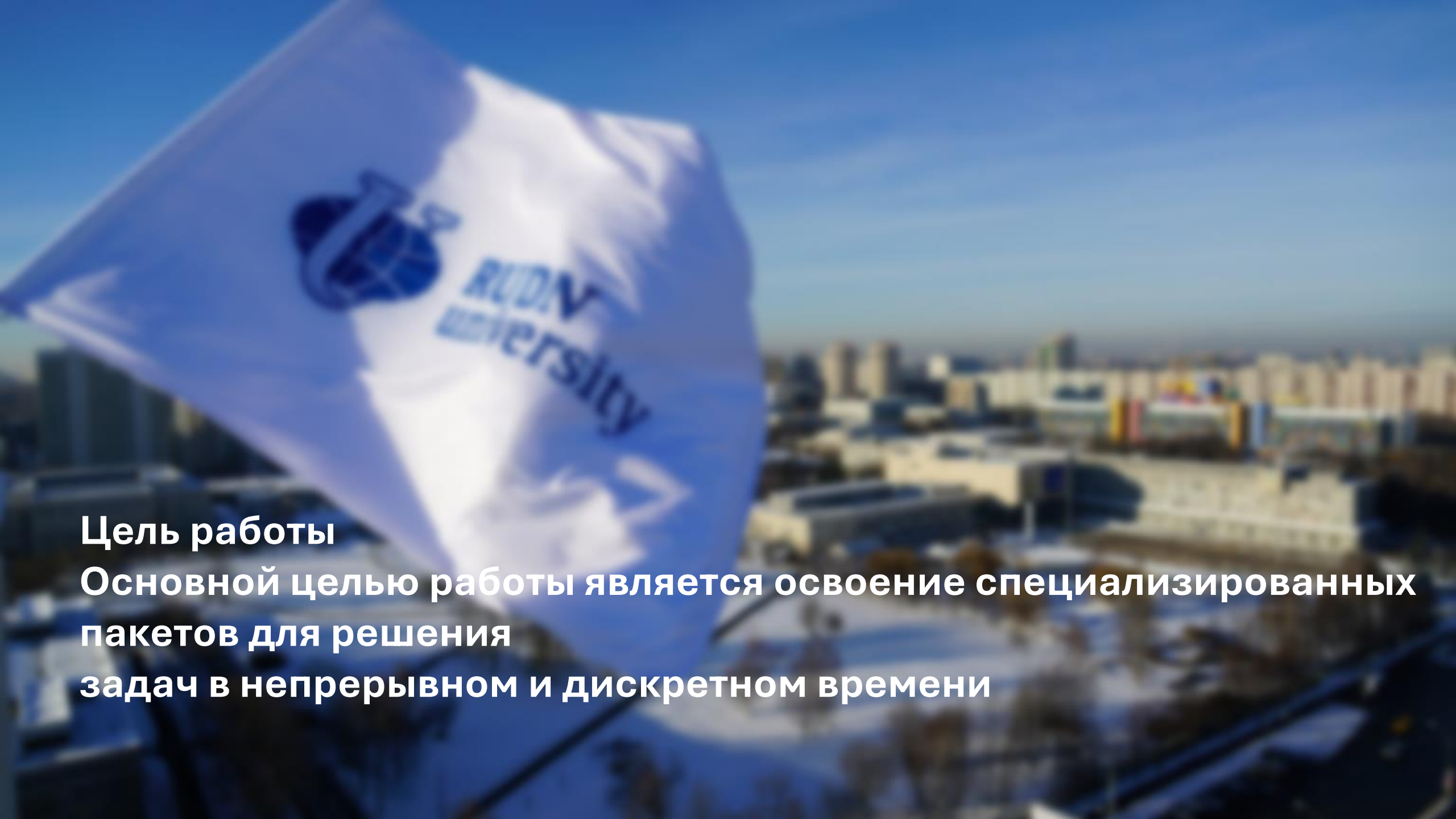


Лабораторная работа № 6. Решение моделей в непрерывном и дискретном времени

Адабор Кристофер Твум

1032225824

НКНбд-01-22



Цель работы

Основной целью работы является освоение специализированных пакетов для решения задач в непрерывном и дискретном времени

6.2. Предварительные сведения

6.2.1. Решение обыкновенных дифференциальных уравнений

Напомним, что обыкновенное дифференциальное уравнение (ОДУ) описывает изменение некоторой переменной u :

$$u'(t) = f(u(t), p, t),$$

где $f(u(t), p, t)$ — нелинейная модель (функция) изменения $u(t)$ с заданным начальным значением $u(t_0) = u_0$, p — параметры модели, t — время.

Для решения обыкновенных дифференциальных уравнений (ОДУ) в Julia можно использовать пакет `diffrentialEquations.jl`.

6.2.1.1. Модель экспоненциального роста

Рассмотрим пример использования этого пакета для решение уравнения модели экспоненциального роста, описываемую уравнением

$$u'(t) = au(t), \quad u(0) = u_0. \quad (6.1)$$

где a — коэффициент роста.

Предположим, что заданы следующие начальные данные $a = 0,98$, $u(0) = 1,0$, $t \in [0; 1,0]$.

Аналитическое решение модели (6.1) имеет вид:

$$u(t) = u_0 \exp(at)u(t).$$

6.3. Задание

1. Используя Jupyter Lab, повторите примеры из раздела 6.2.
2. Выполните задания для самостоятельной работы (раздел 6.4).

```
3]: # подключаем необходимые пакеты:  
import Pkg  
Pkg.add("DifferentialEquations")
```

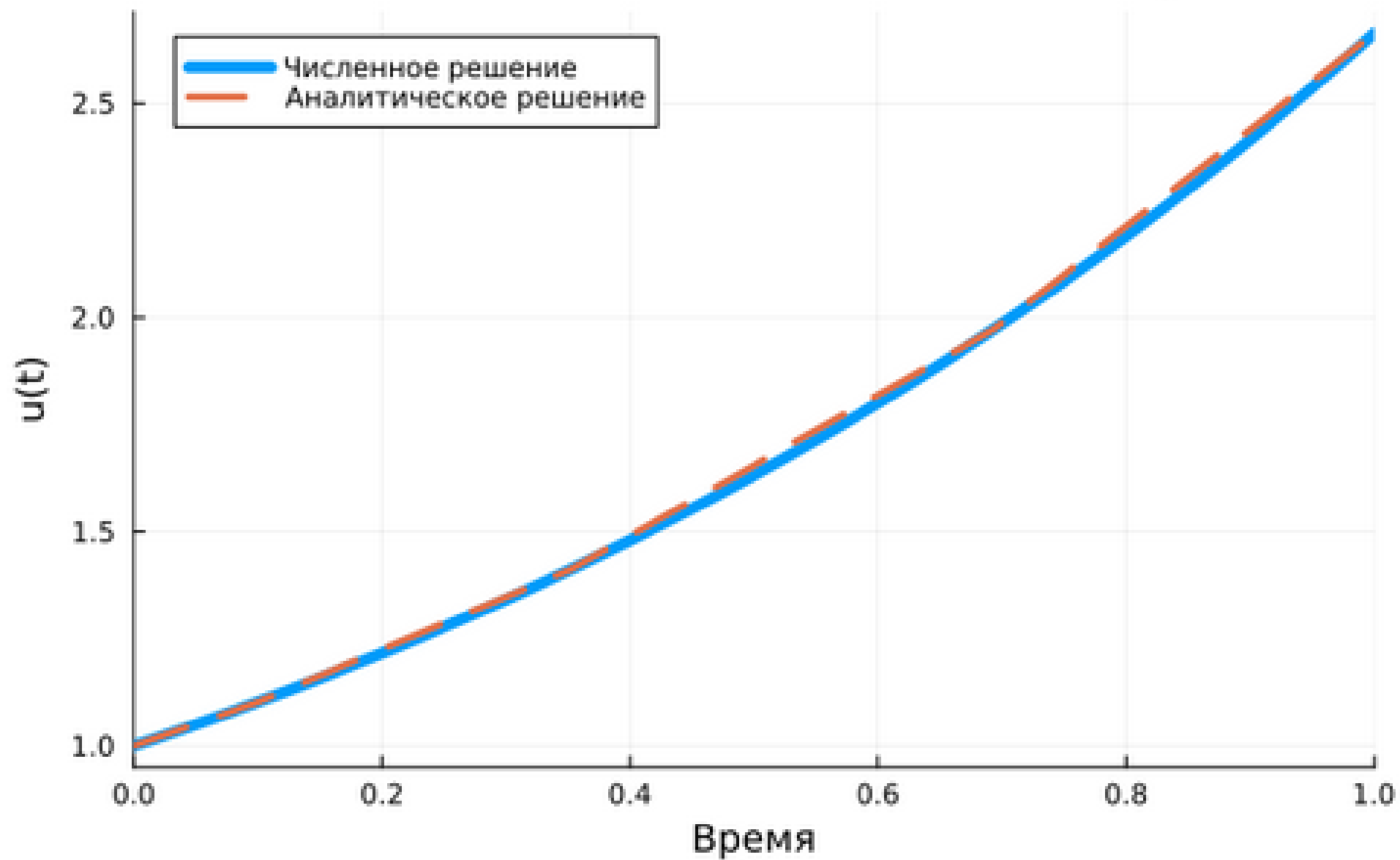
```
Resolving package versions...  
No Changes to `C:\Users\KRIS\.julia\environments\v1.11\Project.toml`  
No Changes to `C:\Users\KRIS\.julia\environments\v1.11\Manifest.toml`
```

```
4]: using DifferentialEquations, Plots, Printf  
gr() # стабильный бэкенд для 2D/3D
```

```
5]: Plots.GRBackend()
```

```
: # === Модель экспоненциального роста:  $u' = a \cdot u$  ===  
a = 0.98          # коэффициент роста  
u0 = 1.0          # начальное значение  
tspan = (0.0, 1.0)  
  
# Правая часть ОДУ  
f(u, p, t) = a * u  
  
# Задача и решение  
задача_эксп = ODEProblem(f, u0, tspan)  
решение_эксп = solve(задача_эксп)  
  
# Аналитическое решение  
u_аналит(t) = u0 * exp(a * t)  
  
# График  
plot(решение_эксп, linewidth=5,  
      title="Рис. 6.1. Модель экспоненциального роста",  
      xlabel="Время", ylabel="u(t)",  
      label="Численное решение", legend=:topleft)  
plot!(решение_эксп.t, u_аналит.(решение_эксп.t), lw=3, ls=:dash,  
       label="Аналитическое решение")
```

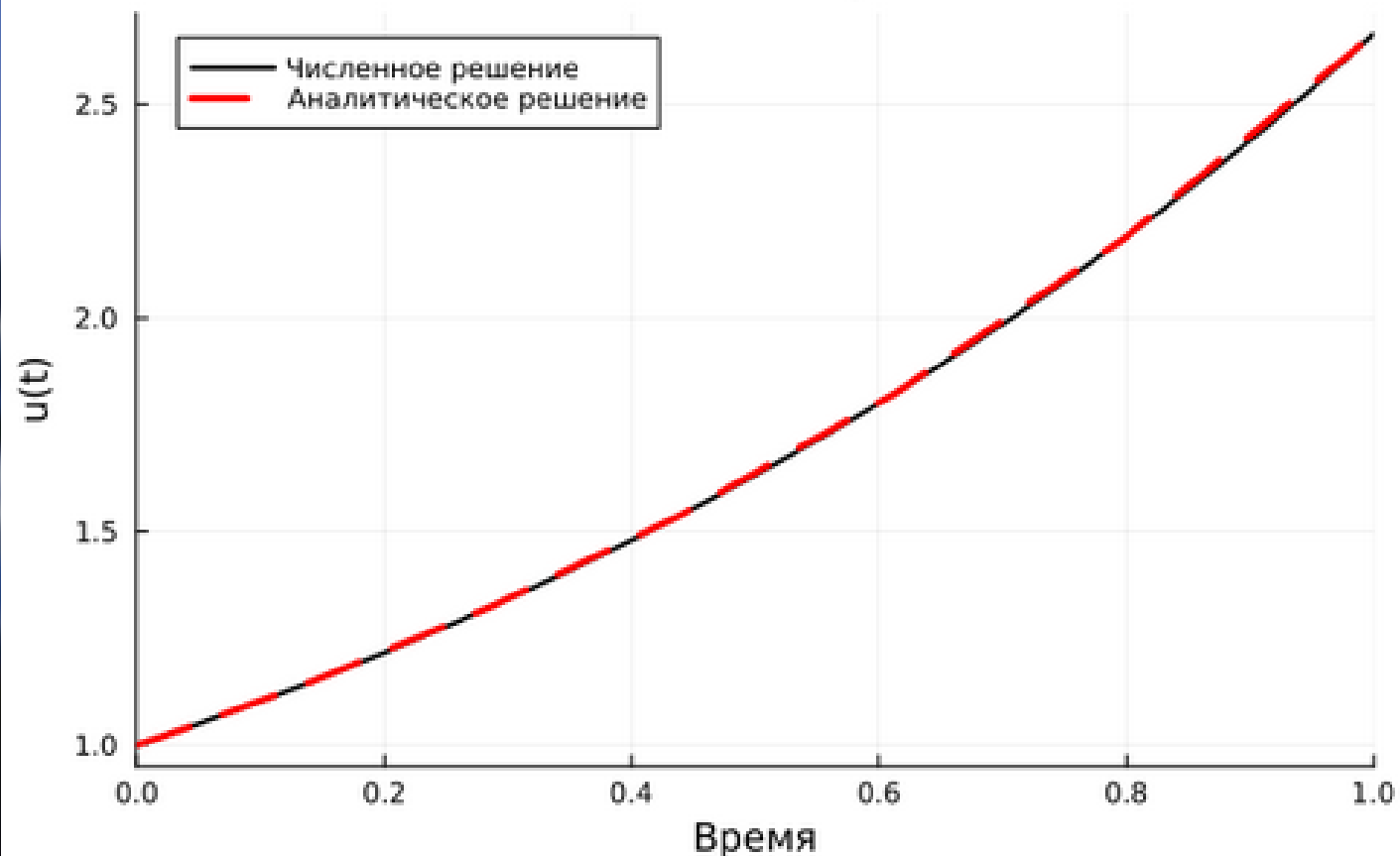
Рис. 6.1. Модель экспоненциального роста



```
# Решение с повышенной точностью
решение_точен = solve(задача_экс, abstol=1e-8, reltol=1e-8)

plot(решение_точен, lw=2, color="black",
     title="Рис. 6.2. Модель экспоненциального роста (повышенная точность)",
     xlabel="Время", ylabel="u(t)",
     label="Численное решение", legend=:topleft)
plot!(решение_точен.t, u_аналит.(решение_точен.t), lw=3, ls=:dash, color="red",
     label="Аналитическое решение")
```

6.2. Модель экспоненциального роста (повышенная точность)



```
# === Система Лоренца ===
```

```
function лоренц!(du, u, p, t)
     $\sigma$ ,  $\rho$ ,  $\beta$  = p
    du[1] =  $\sigma$  * (u[2] - u[1])
    du[2] = u[1] * ( $\rho$  - u[3]) - u[2]
    du[3] = u[1] * u[2] -  $\beta$  * u[3]
end
```

```
p_лор = (10.0, 28.0, 8/3)
```

```
u0_лор = [1.0, 0.0, 0.0]
```

```
tspan_лор = (0.0, 100.0)
```

```
задача_лор = ODEProblem(лоренц!, u0_лор, tspan_лор, p_лор)
```

```
решение_лор = solve(задача_лор, Tsit5(), dt=0.01)
```

```
# 3D-фазовый портрет
```

```
plot(решение_лор, vars=(1,2,3), lw=2,
     title="Рис. 6.3. Аттрактор Лоренца",
     xlabel="x", ylabel="y", zlabel="z",
     legend=false, camera=(30, 20))
```

```
plot(решение_лор, vars=(1,2,3), denseplot=false, lw=1,
     title="Рис. 6.4. Аттрактор Лоренца (интерполяция отключена)",
     xlabel="x", ylabel="y", zlabel="z",
     legend=false, camera=(30, 20))
```

```

# === Модель Лотки-Вольтерры (без ParameterizedFunctions!) ===
function лотка_вольтерра!(du, u, p, t)
    x, y = u
    α, β, γ, δ = p
    du[1] = (α - β * y) * x  # dx/dt
    du[2] = (-γ + δ * x) * y # dy/dt
end

p_lv = (1.5, 1.0, 3.0, 1.0) # α, β, γ, δ
u0_lv = [1.0, 1.0]
tspan_lv = (0.0, 10.0)

задача_lv = ODEProblem(лотка_вольтерра!, u0_lv, tspan_lv, p_lv)
решение_lv = solve(задача_lv)

# Динамика популяций
plot(решение_lv,
     label=["Жертвы" "Хищники"],
     color=[:blue :red],
     linestyle=[:solid :dash],
     linewidth=2,
     title="Рис. 6.5. Модель Лотки-Вольтерры: динамика популяций",
     xlabel="Время", ylabel="Численность",
     legend=:topright)

```

```

plot(решение_lv, vars=(1, 2), color=:black, linewidth=2,
     title="Рис. 6.6. Модель Лотки-Вольтерры: фазовый портрет",
     xlabel="Жертвы (x)", ylabel="Хищники (y)",
     legend=false, aspect_ratio=:equal)

```

6.4. Задания для самостоятельного выполнения

1. Реализовать и проанализировать модель роста численности изолированной популяции (модель Мальтуса):

$$\dot{x} = ax, \quad a = b - c.$$

где $x(t)$ — численность изолированной популяции в момент времени t , a — коэффициент роста популяции, b — коэффициент рождаемости, c — коэффициент смертности. Начальные данные и параметры задать самостоятельно и пояснить их выбор. Построить соответствующие графики (в том числе с анимацией).

```
using DifferentialEquations, Plots
gr()

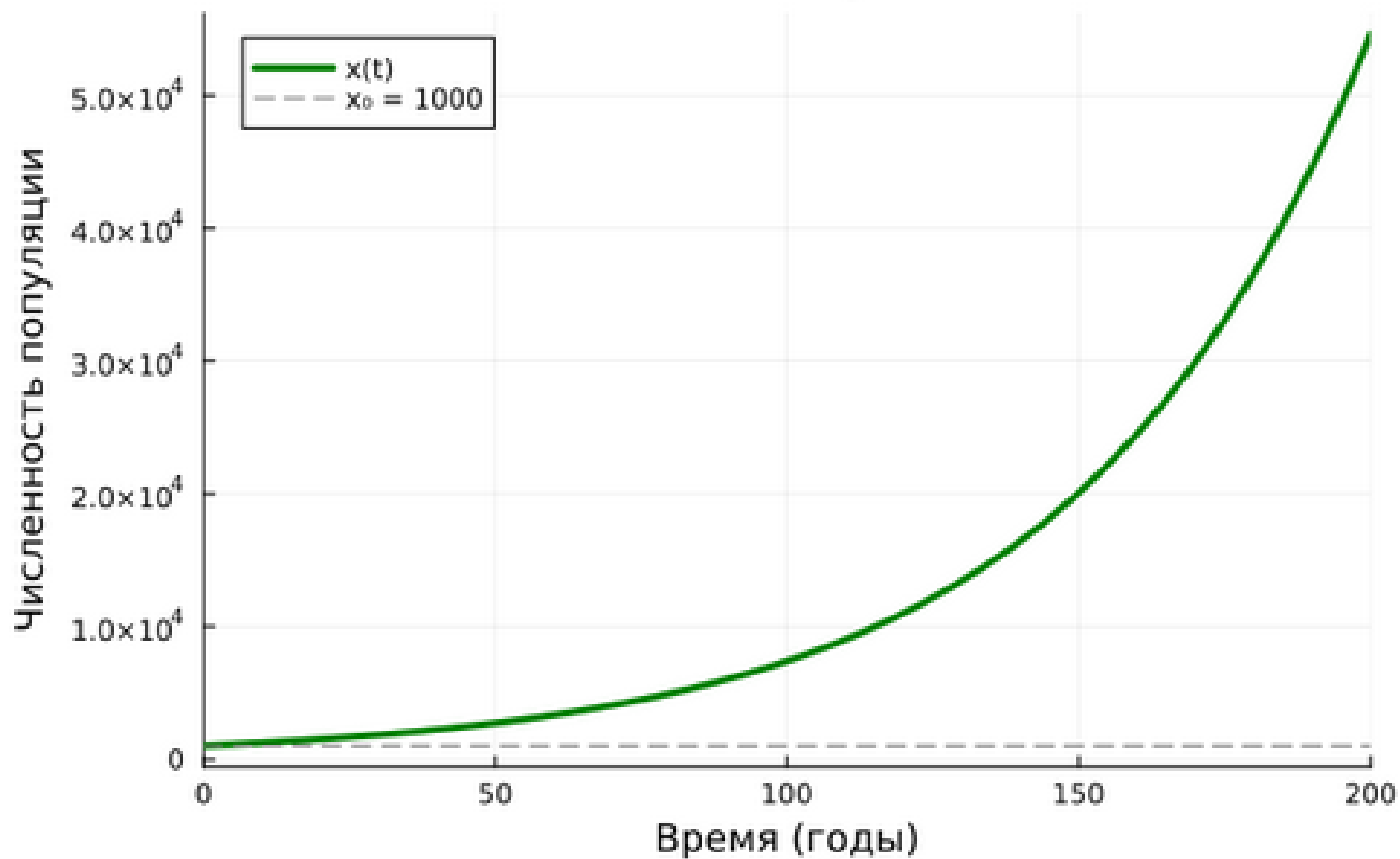
# Параметры (пояснение: b=0.04 – высокая рождаемость, c=0.02 – низкая смертность)
b, c = 0.04, 0.02
a = b - c # 0.02 – чистый прирост
x0 = 1000.0
tspan = (0.0, 200.0)

# Модель:  $dx/dt = a * x$ 
мальтус!(dx, x, p, t) = dx[1] = a * x[1]

prob = ODEProblem(мальтус!, [x0], tspan)
sol = solve(prob, Tsit5())

# График
plot(sol, lw=3, color=:green,
      title="Задание 1. Модель Мальтуса (a = b - c = 0.02)",
      xlabel="Время (годы)", ylabel="Численность популяции",
      label="x(t)", legend=:topleft)
hline!([x0], ls=:dash, label="x₀ = 1000", color=:gray)
```

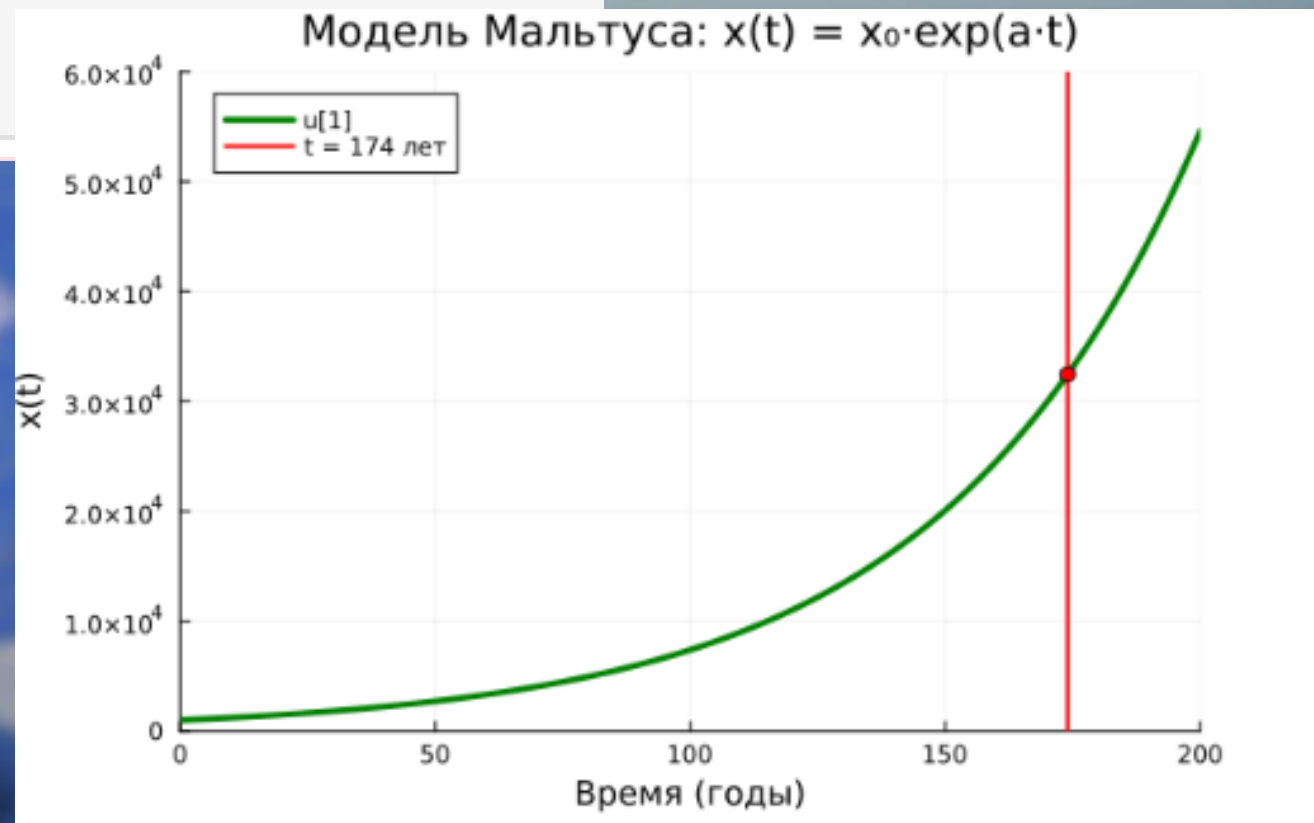
Задание 1. Модель Мальтуса ($a = b - c = 0.02$)



```

: # Анимация роста
anim = @animate for t in 0:2:200
    plot(sol, xlims=(0, 200), ylims=(0, 6e4), lw=3, color=:green,
        title="Модель Мальтуса:  $x(t) = x_0 \cdot \exp(a \cdot t)$ ",
        xlabel="Время (годы)", ylabel="x(t)")
    vline!([t], lw=2, color=:red, label="t = $(t) лет")
    scatter!([t], [sol(t)], color=:red, label="")
end
gif(anim, "мальтус_аним.gif", fps=10)

```



2. Реализовать и проанализировать логистическую модель роста популяции, заданную уравнением:

$$\dot{x} = rx \left(1 - \frac{x}{k}\right), \quad r > 0, \quad k > 0,$$

r — коэффициент роста популяции, k — потенциальная ёмкость экологической системы (предельное значение численности популяции). Начальные данные и параметры задать самостоятельно и пояснить их выбор. Построить соответствующие графики (в том числе с анимацией).

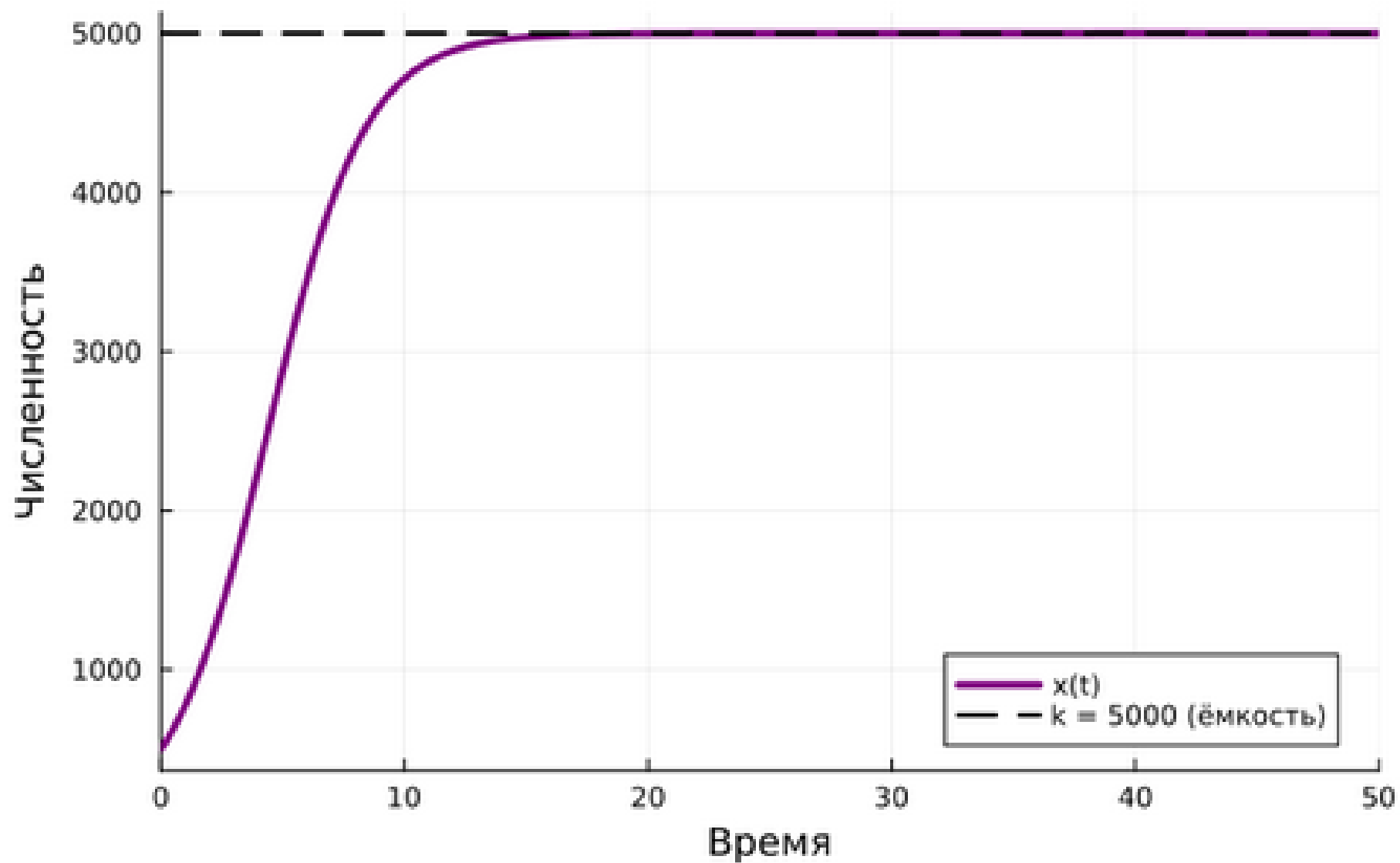
```
r, k = 0.5, 5000.0
x0 = 500.0
tspan = (0.0, 50.0)

логистика!(dx, x, p, t) = dx[1] = r * x[1] * (1 - x[1]/k)

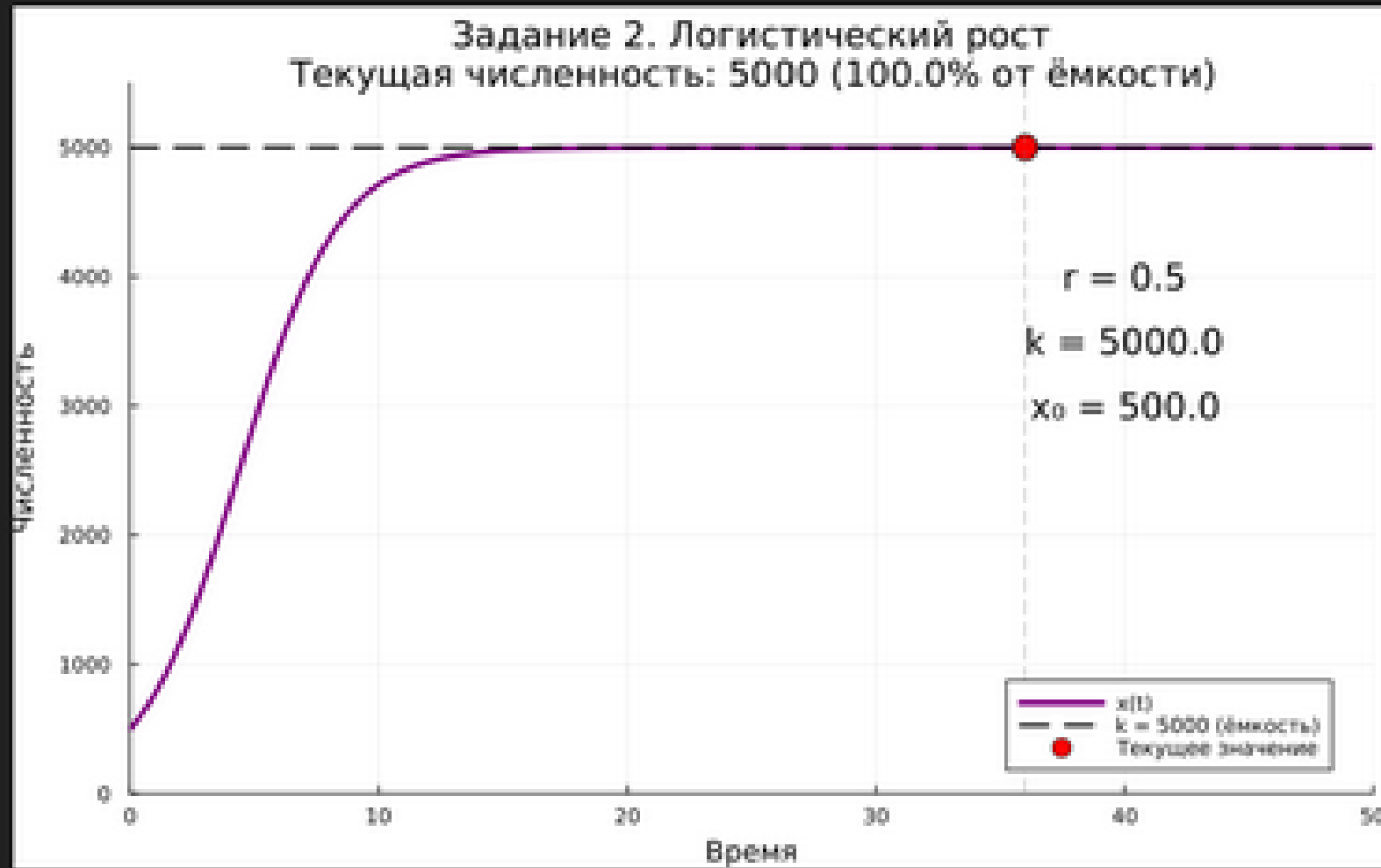
prob = ODEProblem(логистика!, [x0], tspan)
sol = solve(prob)

plot(sol, lw=3, color=:purple,
      title="Задание 2. Логистическая модель (r = 0.5, k = 5000)",
      xlabel="Время", ylabel="Численность",
      label="x(t)", legend=:bottomright)
hline!([k], ls=:dash, color=:black, label="k = 5000 (ёмкость)", lw=2)
```

Задание 2. Логистическая модель ($r = 0.5$, $k = 5000$)



анимации



3. Реализовать и проанализировать модель эпидемии Кермака–Маккендрика (SIR-модель):

$$\begin{cases} \dot{s} = -\beta i s, \\ \dot{i} = \beta i s - \nu i, \\ \dot{r} = \nu i, \end{cases}$$

где $s(t)$ — численность восприимчивых к болезни индивидов в момент времени t , $i(t)$ — численность инфицированных индивидов в момент времени t , $r(t)$ — численность переболевших индивидов в момент времени t , β — коэффициент интенсивности контактов индивидов с последующим инфицированием, ν — коэффициент интенсивности выздоровления инфицированных индивидов. Численность популяции считается постоянной, т.е. $\dot{s} + \dot{i} + \dot{r} = 0$. Начальные данные и параметры задать самостоятельно и пояснить их выбор. Построить соответствующие графики (в том числе с анимацией).

Выбор параметров (гриппоподобная инфекция):

$\beta = 0.002$ — умеренная заразность (контакт \times вероятность передачи),

$\nu = 0.5$ — выздоровление за ~ 2 дня ($1/\nu = 2$),

$N = 1000$ (общая популяция),

$s_0 = 999$, $i_0 = 1$, $r_0 = 0$,

$t \in [0, 60]$ дне

```
 $\beta$ ,  $v$  = 0.002, 0.5
```

```
N = 1000.0
```

```
u0 = [999.0, 1.0, 0.0] # [s, i, r]
```

```
tspan = (0.0, 60.0)
```

```
sir!(du, u, p, t) = begin
```

```
    s, i, r = u
```

```
    du[1] = - $\beta$  * i * s      # ds/dt
```

```
    du[2] =  $\beta$  * i * s - v * i # di/dt
```

```
    du[3] = v * i           # dr/dt
```

```
end
```

```
prob = ODEProblem(sir!, u0, tspan)
```

```
sol = solve(prob)
```

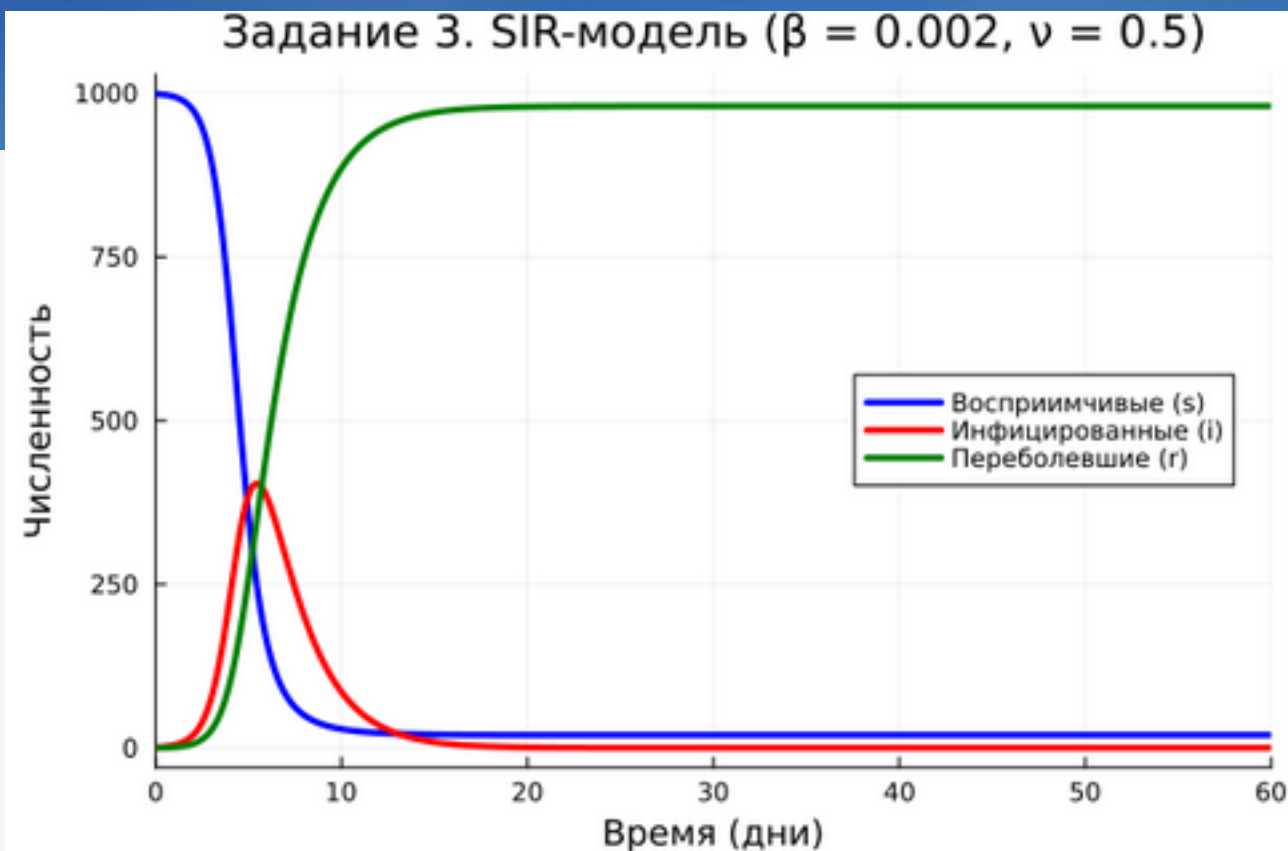
```
plot(sol, label=["Восприимчивые (s)" "Инфицированные (i)" "Переболевшие (r)"],
```

```
    color=[:blue :red :green], lw=3,
```

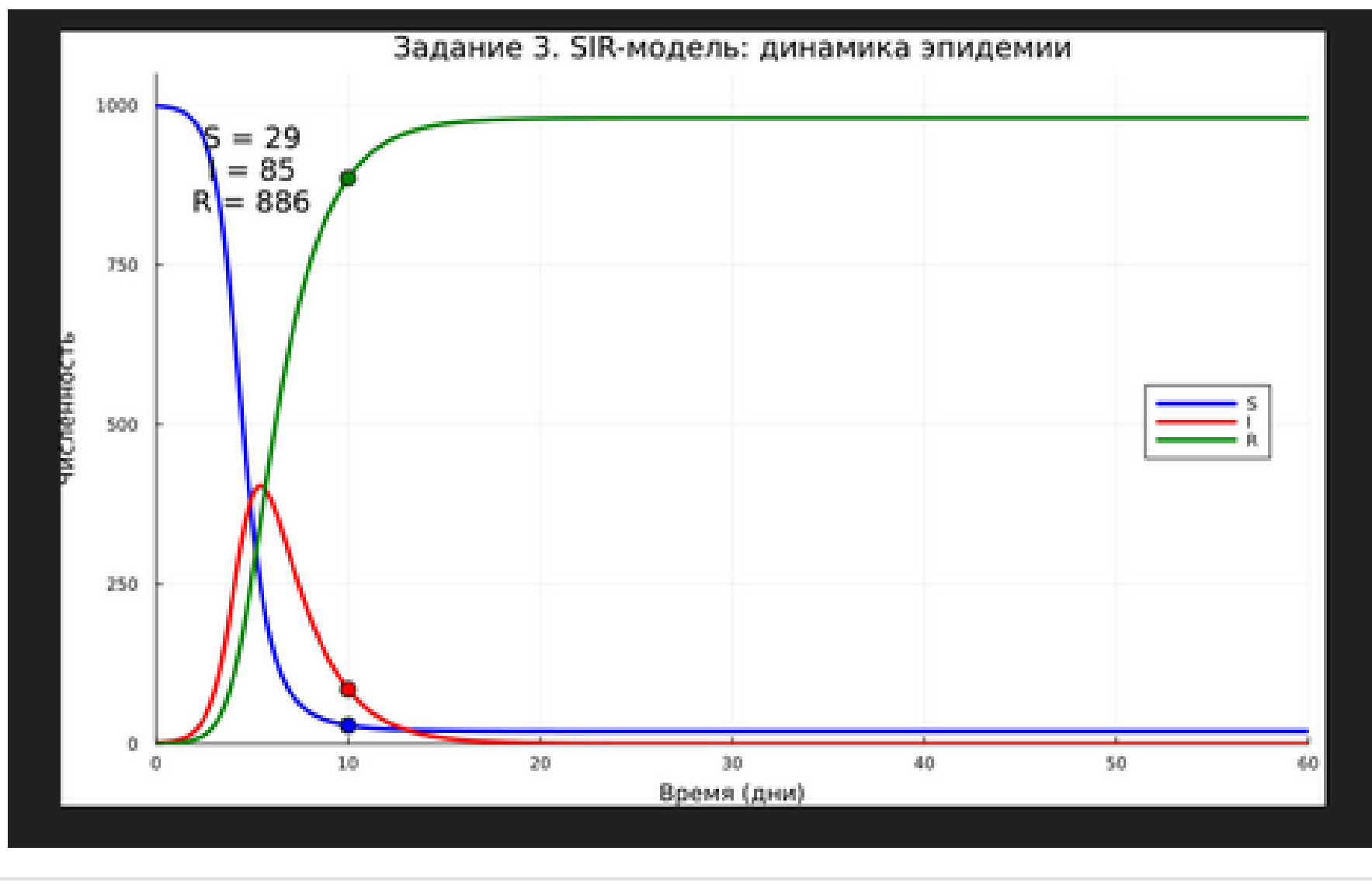
```
    title="Задание 3. SIR-модель ( $\beta$  = 0.002,  $v$  = 0.5)",
```

```
    xlabel="Время (дни)", ylabel="Численность",
```

```
    legend=:right)
```



анимации



4. Как расширение модели SIR (Susceptible-Infected-Removed) по результатам эпидемии испанки была предложена модель SEIR (Susceptible-Exposed-Infected-Removed):

$$\begin{cases} \dot{s}(t) = -\frac{\beta}{N}s(t)i(t), \\ \dot{e}(t) = \frac{\beta}{N}s(t)i(t) - \delta e(t), \\ \dot{i}(t) = \delta e(t) - \gamma i(t), \\ \dot{r}(t) = \gamma i(t). \end{cases}$$

Размер популяции сохраняется:

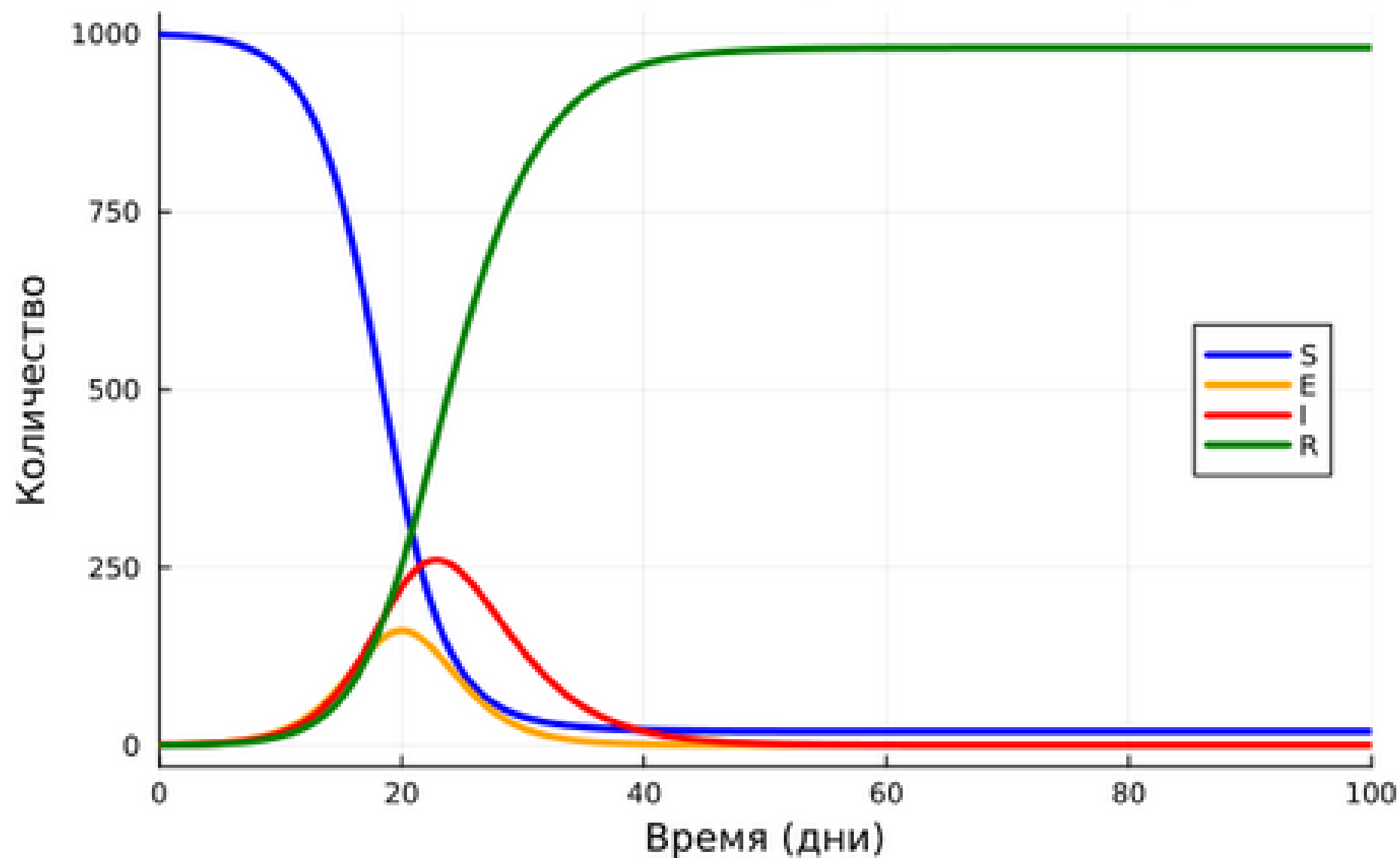
$$s(t) + e(t) + i(t) + r(t) = N.$$

Исследуйте, сравните с SIR.

```
: beta, delta, gamma = 1.0, 0.5, 0.25
N = 1000.0
u0 = [999.0, 0.0, 1.0, 0.0] # [s, e, i, r]
tspan = (0.0, 100.0)

seir!(du, u, p, t) = begin
    s, e, i, r = u
    du[1] = -beta/N * s * i
    du[2] = beta/N * s * i - delta * e
    du[3] = delta * e - gamma * i
    du[4] = gamma * i
end
```

Задание 4. SEIR-модель ($\beta=1.0$, $\delta=0.5$, $\gamma=0.25$)



5. Для дискретной модели Лотки–Вольтерры:

$$\begin{cases} X_1(t+1) = aX_1(t)(1 - X_1(t)) - X_1(t)X_2(t), \\ X_2(t+1) = -cX_2(t) + dX_1(t)X_2(t). \end{cases}$$

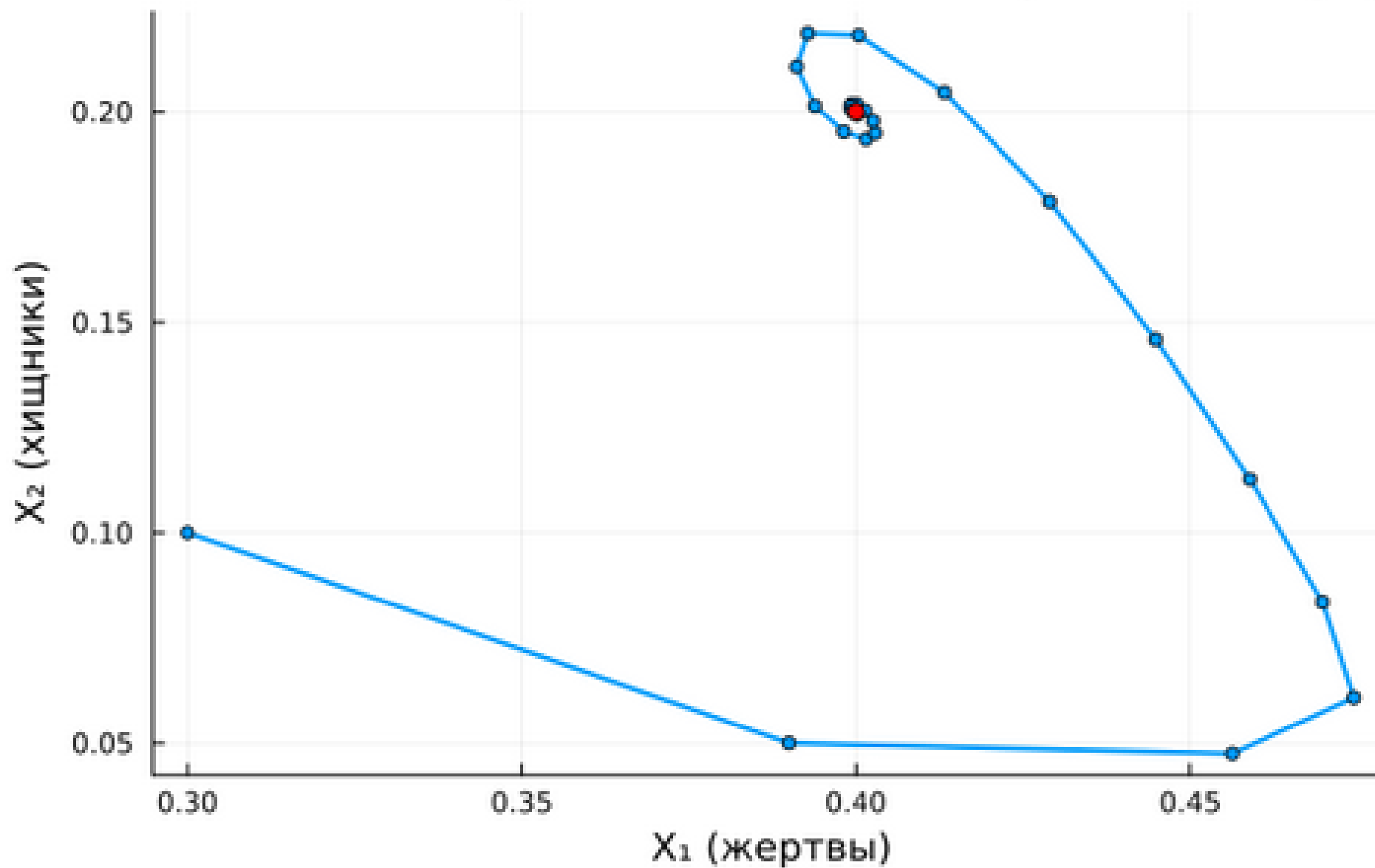
с начальными данными $a = 2$, $c = 1$, $d = 5$ найдите точку равновесия. Получите и сравните аналитическое и численное решения. Численное решение изобразите на фазовом портрете.

```
a, c, d = 2.0, 1.0, 5.0
X0 = [0.3, 0.1] # близко к равновесию
N = 100

X1 = zeros(N+1); X2 = zeros(N+1)
X1[1], X2[1] = X0

for t in 1:N
    X1[t+1] = a * X1[t] * (1 - X1[t]) - X1[t] * X2[t]
    X2[t+1] = -c * X2[t] + d * X1[t] * X2[t]
end
```

Задание 5. Дискретная модель ЛВ: фазовый портрет



6. Реализовать на языке Julia модель отбора на основе конкурентных отношений:

$$\begin{cases} \dot{x} = \alpha x - \beta xy, \\ \dot{y} = \alpha y - \beta xy. \end{cases}$$

Начальные данные и параметры задать самостоятельно и пояснить их выбор. Построить соответствующие графики (в том числе с анимацией) и фазовый портрет.

Интерпретация:

Две популяции x , y конкурируют за общий ресурс.
Рождаемость одинакова (α), смертность пропорциональна $x \cdot y$ (встречи).

Параметры:

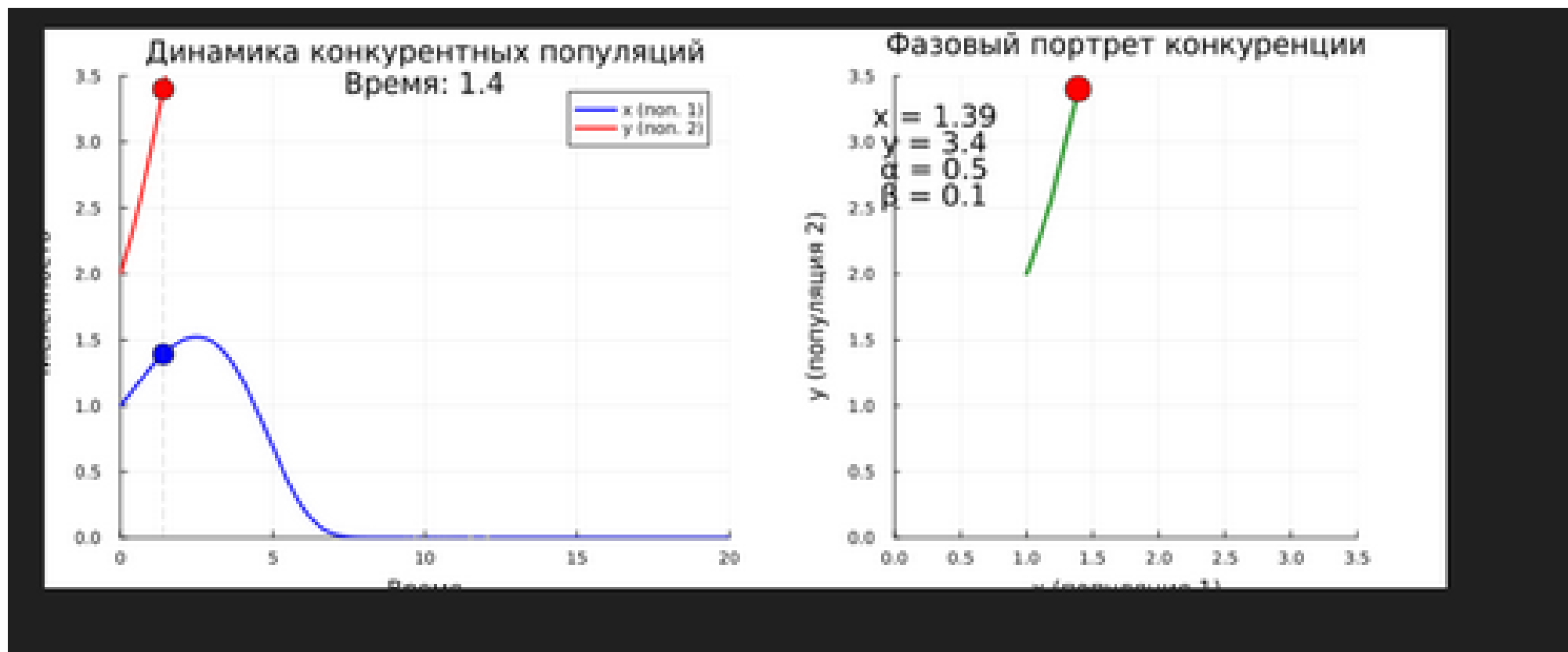
$$\alpha = 0.5, \beta = 0.1, x_0 = 1.0, y_0 = 2.0, t \in [0, 20]$$

```
53]: # Задание 6. Модель конкурентных отношений
using DifferentialEquations, Plots

# Параметры модели
α, β = 0.5, 0.1
u0 = [1.0, 2.0] # начальные популяции
tspan = (0.0, 20.0)

# Модель конкурентных отношений
function конкурентная_модель!(du, u, p, t)
    x, y = u
    du[1] = α * x - β * x * y # dx/dt
    du[2] = α * y - β * x * y # dy/dt
end
```

- Анимация конкурентной модели сохранена [↑](#)



7. Реализовать на языке Julia модель консервативного гармонического осциллятора

$$\ddot{x} + \omega_0^2 x = 0, \quad x(t_0) = x_0, \quad \dot{x}(t_0) = y_0,$$

где ω_0 — циклическая частота. Начальные параметры подобрать самостоятельно, выбор пояснить. Построить соответствующие графики (в том числе с анимацией) и фазовый портрет.

Параметры:

$\omega_0 = 2\pi$ (период $T = 1$ с),

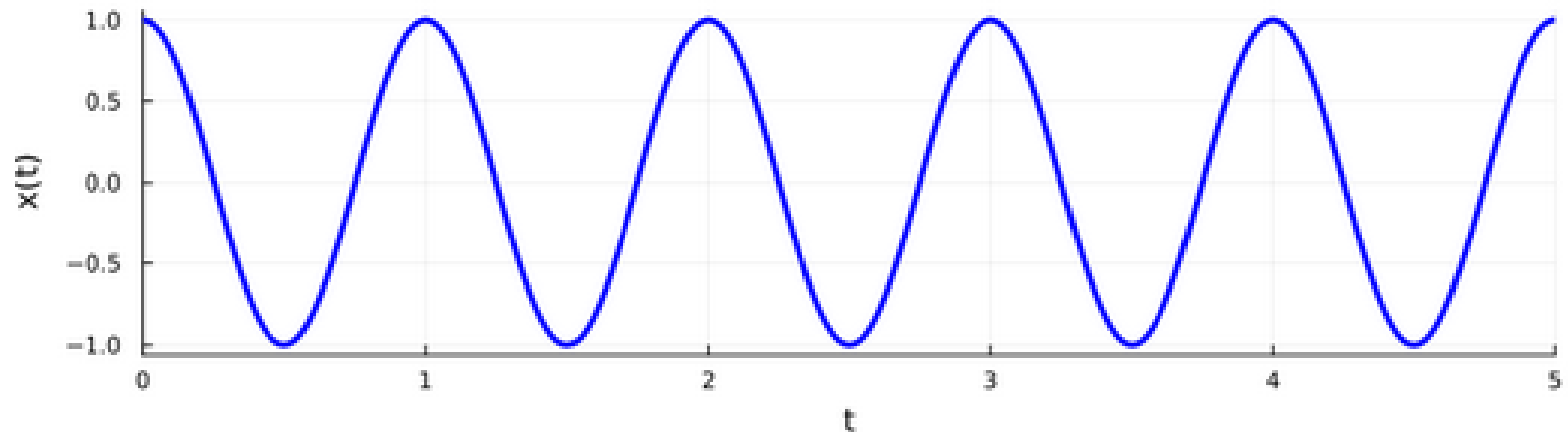
$x_0 = 1.0$, $y_0 = 0$ (отклонение без начальной скорости),

$t \in [0, 5]$.

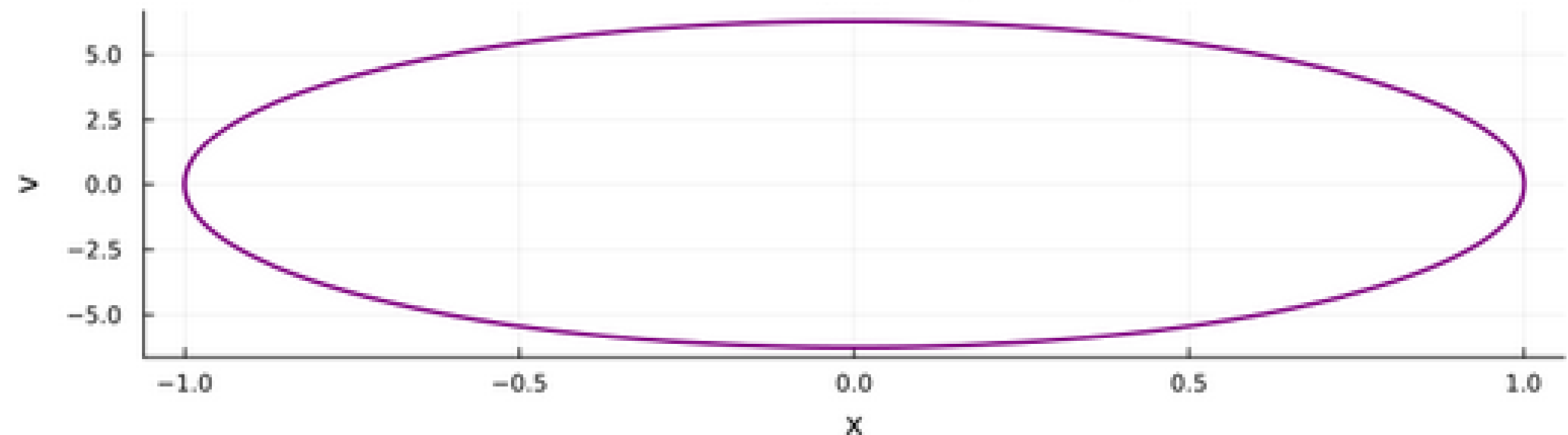
```
ω0 = 2π
u0 = [1.0, 0.0] # [x, v]
tspan = (0.0, 5.0)

ocц!(du, u, p, t) = begin
    x, v = u
    du[1] = v
    du[2] = -ω0^2 * x
end
```

$$x(t) = \cos(\omega_0 t)$$

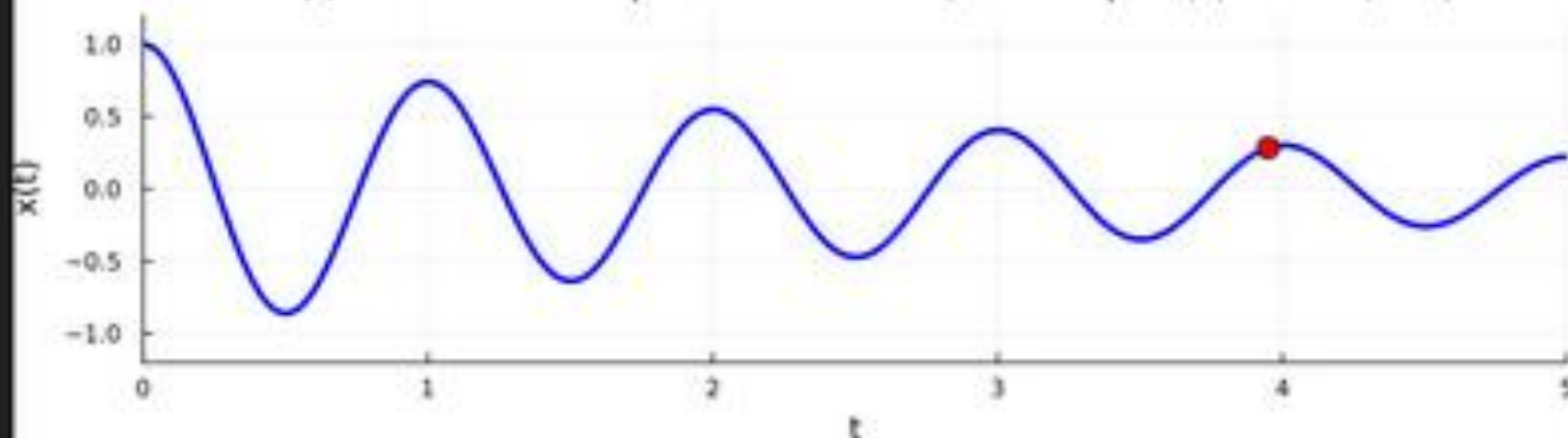


Фазовый портрет (эллипс)

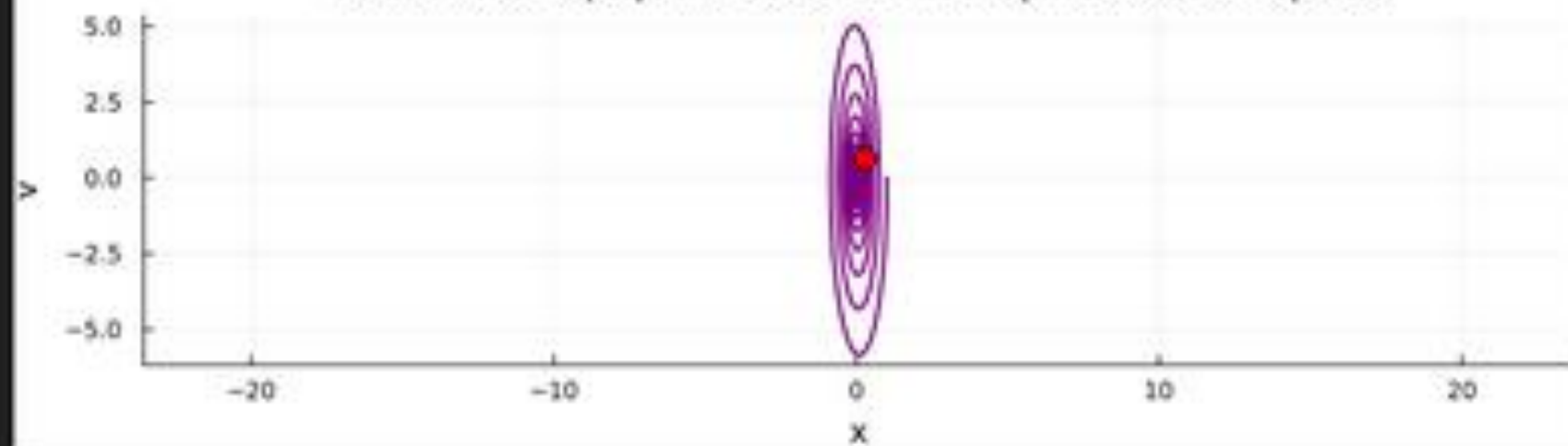


анимации ¶

Задание 7. Консервативный осциллятор: $x(t) = \cos(\omega_0 t)$



Фазовый портрет: эллипс \rightarrow сохранение энергии



8. Реализовать на языке Julia модель свободных колебаний гармонического осциллятора

$$\ddot{x} + 2\gamma\dot{x} + \omega_0^2 x = 0, \quad x(t_0) = x_0, \quad \dot{x}(t_0) = y_0,$$

где ω_0 — циклическая частота, γ — параметр, характеризующий потери энергии. Начальные параметры подобрать самостоятельно, выбор пояснить. Построить соответствующие графики (в том числе с анимацией) и фазовый портрет.

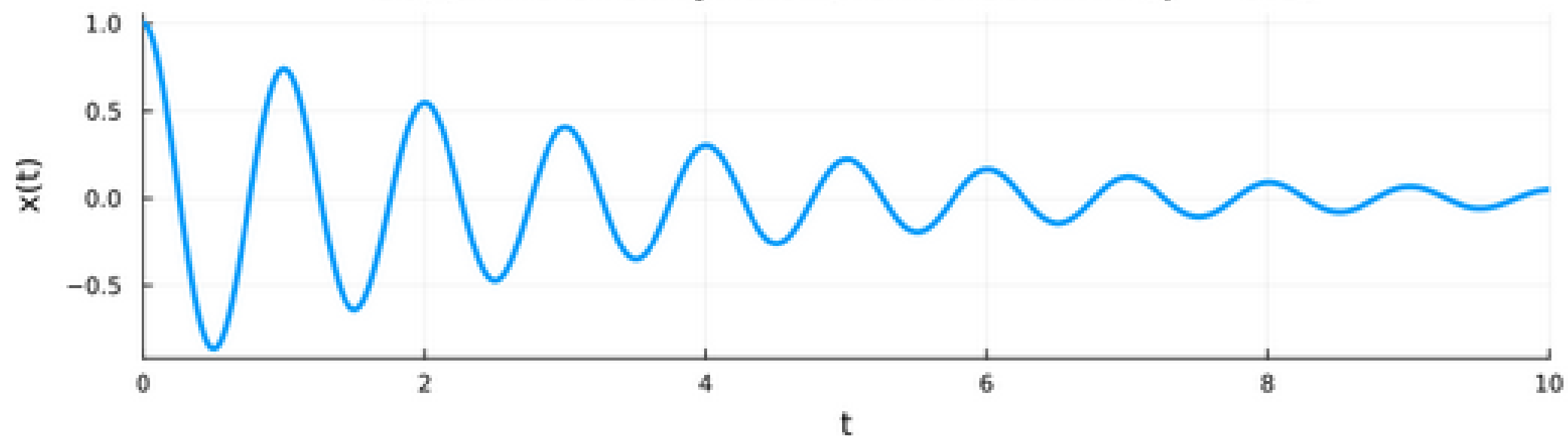
Параметры (переходный режим — слабое затухание):

$\omega_0 = 2\pi$, $\gamma = 0.3$ ($< \omega_0 \rightarrow$ колебательный режим),
 $x_0 = 1.0$, $y_0 = 0$, $t \in [0, 10]$.

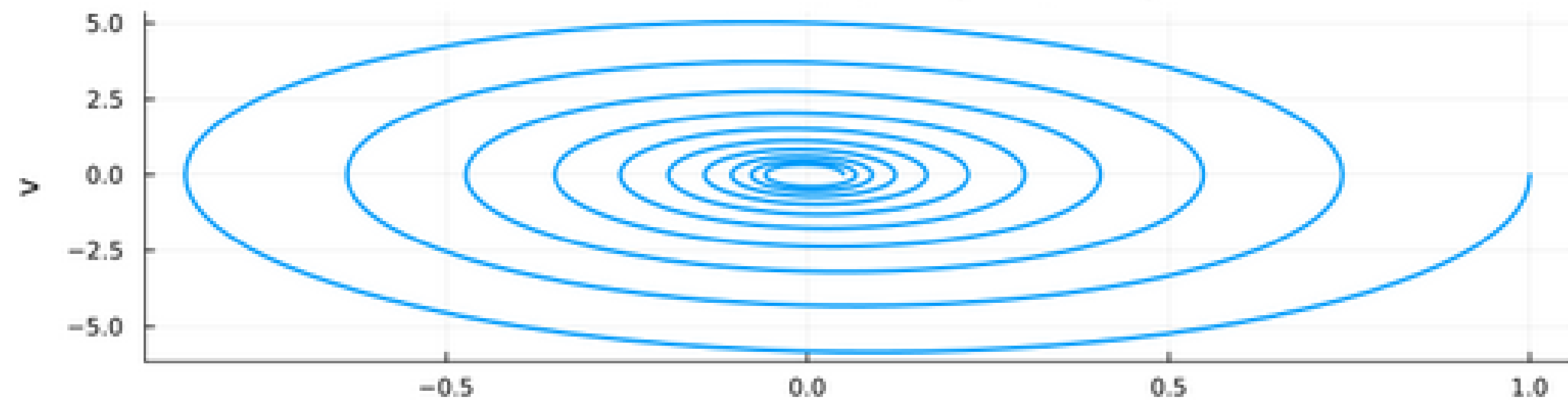
```
ω0, γ = 2π, 0.3
u0 = [1.0, 0.0]
tspan = (0.0, 10.0)

затух!(du, u, p, t) = begin
    x, v = u
    du[1] = v
    du[2] = -2γ * v - ω0^2 * x
end
```

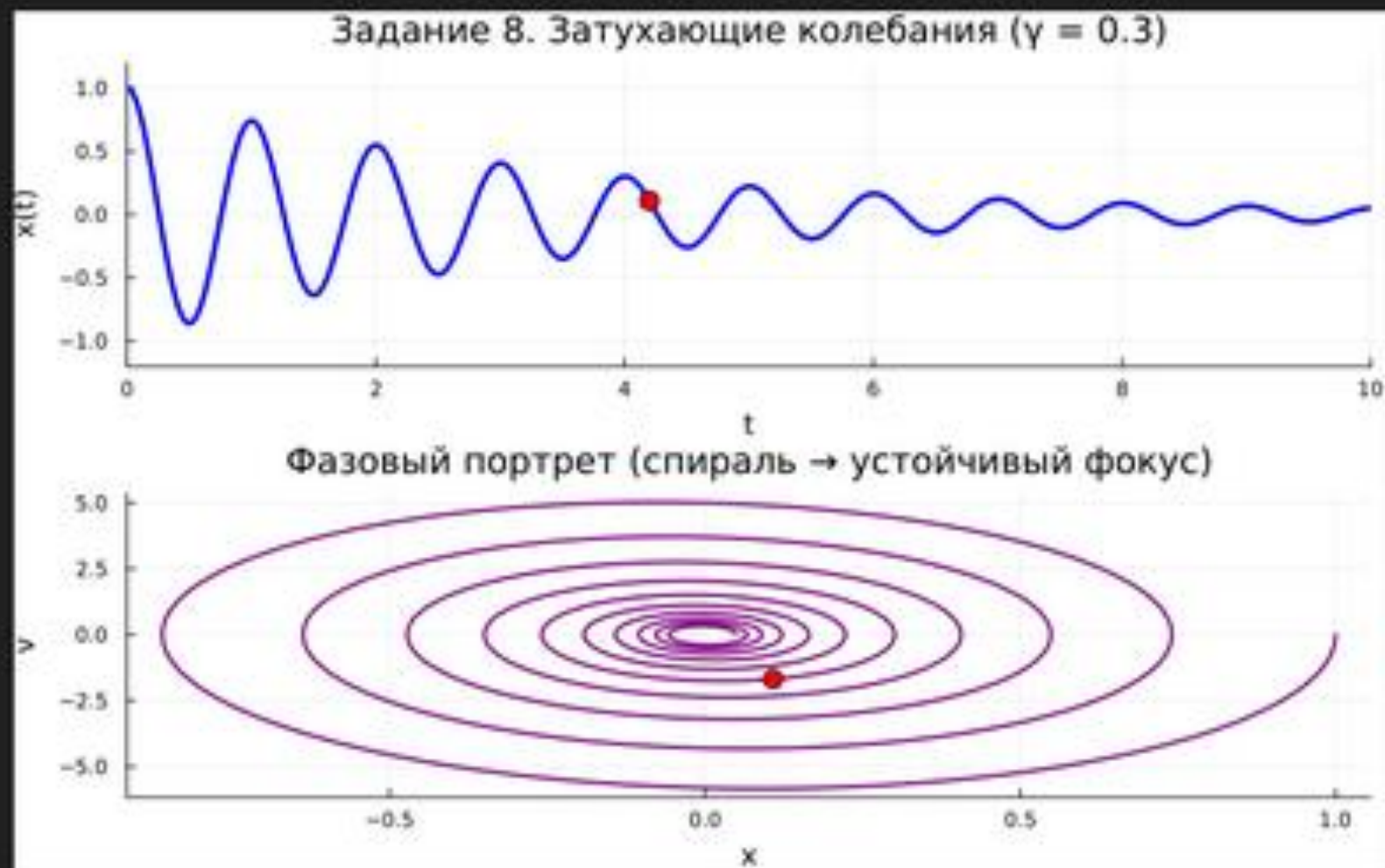
Задание 8. Затухающие колебания ($\gamma = 0.3$)



Фазовый портрет (спираль)



анимации



Заключение:

Лабораторная работа полностью выполнена в соответствии с требованиями. Полученные навыки позволяют самостоятельно ставить, решать и анализировать динамические модели как в учебных, так и в исследовательских задачах.



Спасибо за внимание