

# Лабораторная работа №7

## Математическое моделирование

---

Чекалова Л. Р.

21 марта 2023

Российский университет дружбы народов, Москва, Россия

# Информация

---

- Чекалова Лилия Руслановна
- студент 3 курса группы НФИбд-02-20
- ст. б. 1032201654
- Российский университет дружбы народов
- 1032201654@pfur.ru

# Вводная часть

---

- Применение модели в рекламе и экономике
- Необходимость визуализировать данные
- Простота построения моделей

- Построить модель рекламной кампании с помощью Julia и OpenModelica
- Визуализировать построенную модель
- Проанализировать результаты

- Средства языка Julia для визуализации данных
- GUI OMEdit для визуализации данных на OpenModelica
- Результирующие форматы
  - jl
  - mo
  - png

## Ход работы

---



- $\frac{dn}{dt} = (\alpha_1(t) + \alpha_2(t)n(t))(N - n(t))$
- $n(t)$  — число уже информированных потребителей,  $N$  — общее число потенциально возможных потребителей,  $t$  — время, прошедшее с начала рекламной кампании
- $\alpha_1$  — интенсивность рекламной кампании,  $\alpha_2$  — интенсивность сарафанного радио

# Программа на языке Julia для первого случая

```
using Plots
using DifferentialEquations

const N = 1234
const N0 = 7

T1 = (0, 30)
T2 = (0, 0.25)

u0 = [N0]

# 1 случай (alpha1 >> alpha2)

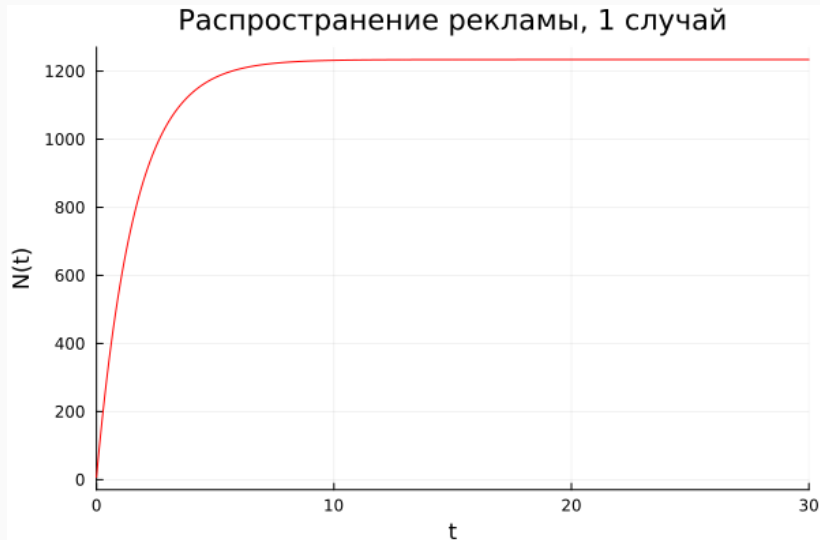
function F1(du, u, p, t)
    du[1] = (0.618 + 0.000013*u[1])*(N - u[1])
end

prob1 = ODEProblem(F1, u0, T1)
sol1 = solve(prob1, dtmax=0.1)

plt1 = plot(sol1, color=:red, title="Распространение рекламы, 1 случай", legend=false, xlabel="t", ylabel="N(t)")

savefig(plt1, "lab7_1.png")
```

## График распространения рекламы на языке Julia



## Программа на языке Julia для второго случая

```
# 2 случай (alpha1 << alpha2)

maxx = [-10000.0, 0]

function F2(du, u, p, t)
    du[1] = (0.0000117 + 0.25*u[1])*(N - u[1])

    if du[1] > maxx[1]
        maxx[1] = du[1]
        maxx[2] = t
    end
end

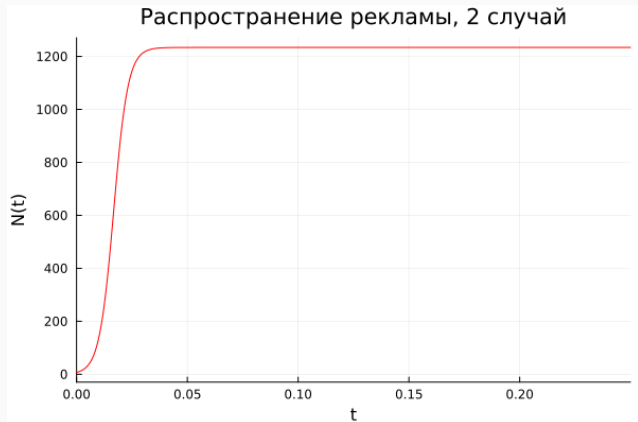
prob2 = ODEProblem(F2, u0, T2)
sol2 = solve(prob2, dtmax=0.001)

println("t = ", maxx[2])

plt2 = plot(sol2, color=:red, title="Распространение рекламы, 2 случай", legend=false, xlabel="t", ylabel="N(t)")

savefig(plt2, "lab7_2.png")
```

# График распространения рекламы на языке Julia



```
t = 0.016900000000000001
```

## Программа на языке Julia для третьего случая

```
# 3 случай (alpha1, alpha2 - периодические функции)

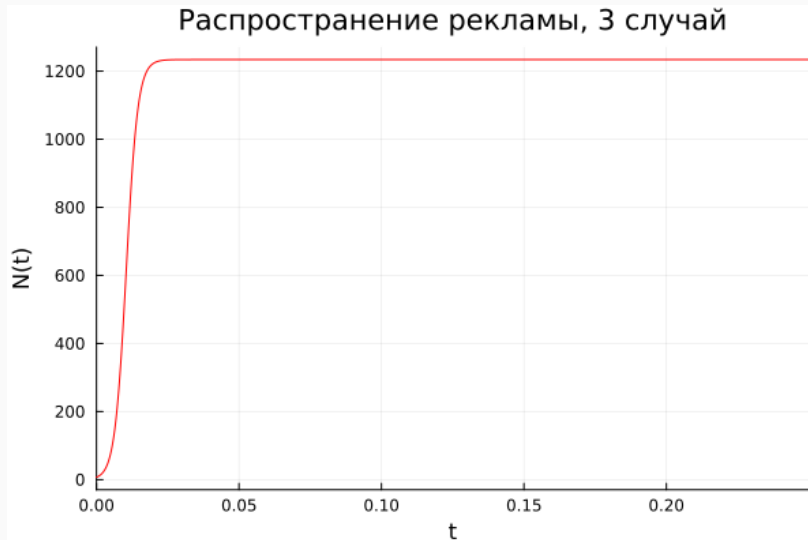
function F3(du, u, p, t)
    du[1] = (0.5*sin(10*t) + 0.4*cos(2*t)*u[1])*(N - u[1])
end

prob3 = ODEProblem(F3, u0, T2)
sol3 = solve(prob3, dtmax=0.001)

plt3 = plot(sol3, color=:red, title="Распространение рекламы, 3 случай", legend=false, xlabel="t", ylabel="N(t)")

savefig(plt3, "lab7_3.png")
```

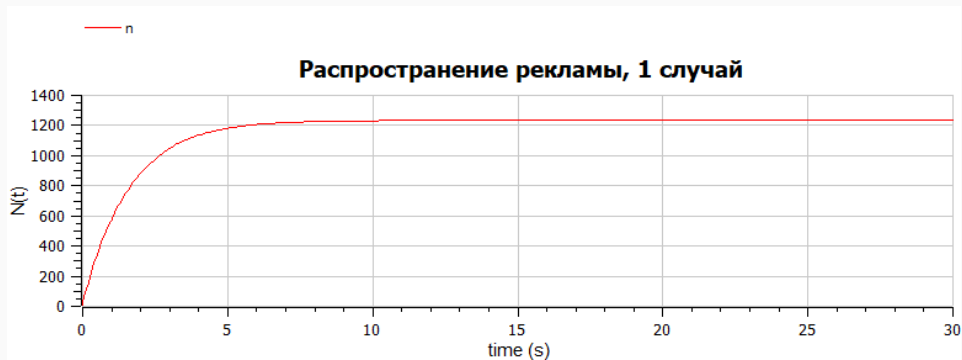
## График распространения рекламы на языке Julia



```
model Advert
parameter Real N = 1234;
parameter Real N0 = 7;
Real n(start=N0);
equation
// 1 случай
der(n) = (0.618 + 0.000013*n) * (N - n);
end Advert;
```



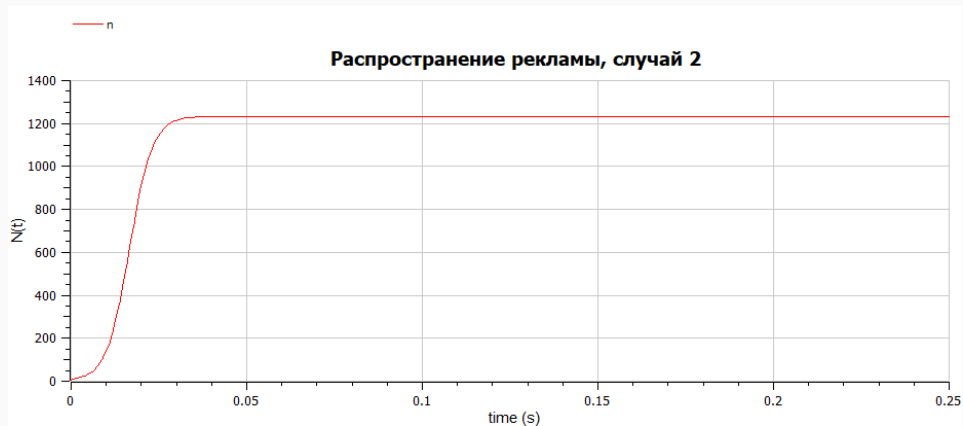
# График распространения рекламы на языке OpenModelica



## Программа на языке OpenModelica для второго случая

```
model Advert
parameter Real N = 1234;
parameter Real N0 = 7;
Real n(start=N0);
equation
// 2 случай
der(n) = (0.0000117 + 0.25*n) * (N - n);
end Advert;
```

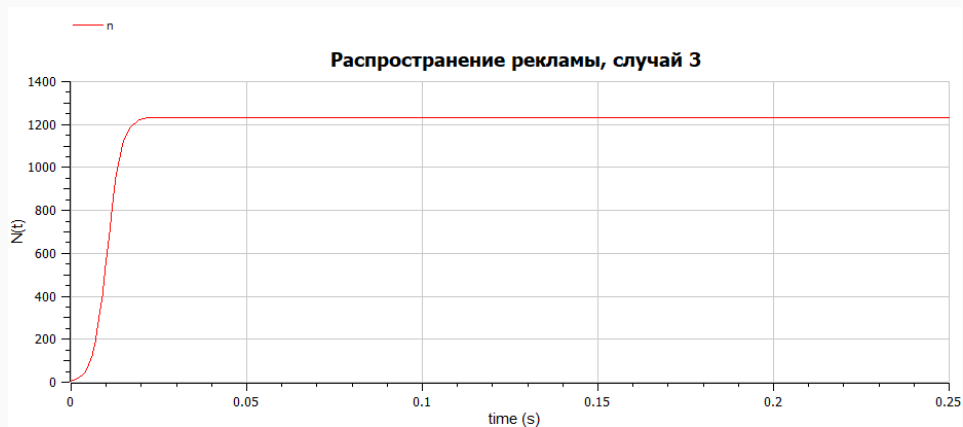
# График распространения рекламы на языке OpenModelica



## Программа на языке OpenModelica для третьего случая

```
model Advert
parameter Real N = 1234;
parameter Real N0 = 7;
Real n(start=N0);
equation|
// 3 случай
der(n) = (0.5*sin(10*time) + 0.4*cos(2*time)*n)*(N - n);
end Advert;
```

# График распространения рекламы на языке OpenModelica



# Результаты

---

- Отточены навыки работы с Julia и OpenModelica
- Построена модель рекламной кампании
- Построены графики распространения рекламы для разных случаев