

Лабораторная работа №7

Математическое моделирование

Байрамгельдыев Довлетмурат

25 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Байрамгельдыев Довлетмурат
- студент 3 курса группы НФИбд-01-20
- ст. б. 1032207470
- Российский университет дружбы народов
- 1032207470@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
- α_1 — интенсивность рекламной кампании, α_2 — интенсивность сарафанного радио

Программа на языке Julia для первого случая

```
using Plots
using DifferentialEquations

const N = 500
const N0 = 5

T1 = (0, 30)
T2 = (0, 0.2)

u0 = [N0]

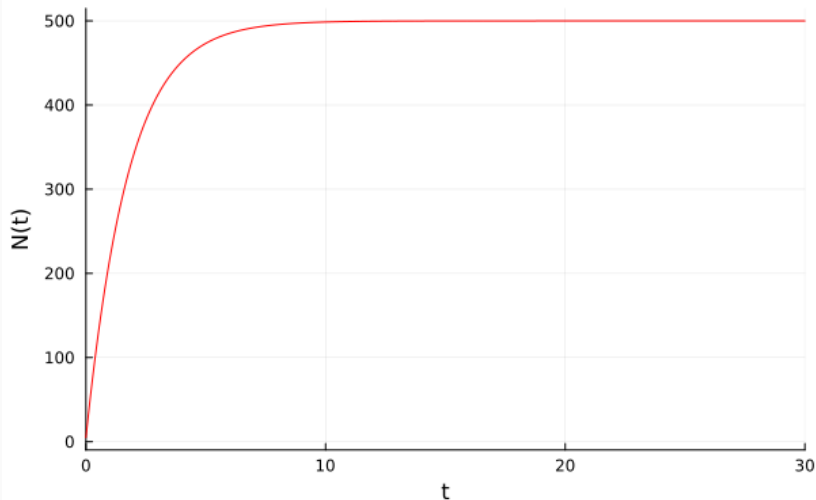
# 1 случай (alpha1 >> alpha2)

function F1(du, u, p, t)
    du[1] = (0.55 + 0.0001*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")
```

Распространение рекламы, 1 случай



Программа на языке Julia для второго случая

```
# 2 случай (alpha1 << alpha2)

maxx = [-10000.0, 0]

function F2(du, u, p, t)
    du[1] = (0.00005 + 0.2*u[1])*(N - u[1])

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

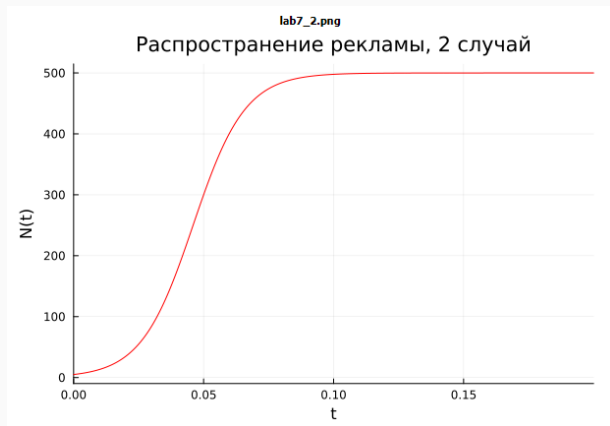
prob2 = ODEProblem(F2, u0, T2)
sol2 = solve(prob2, dtmax=0.001)

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



Программа на языке Julia для третьего случая

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

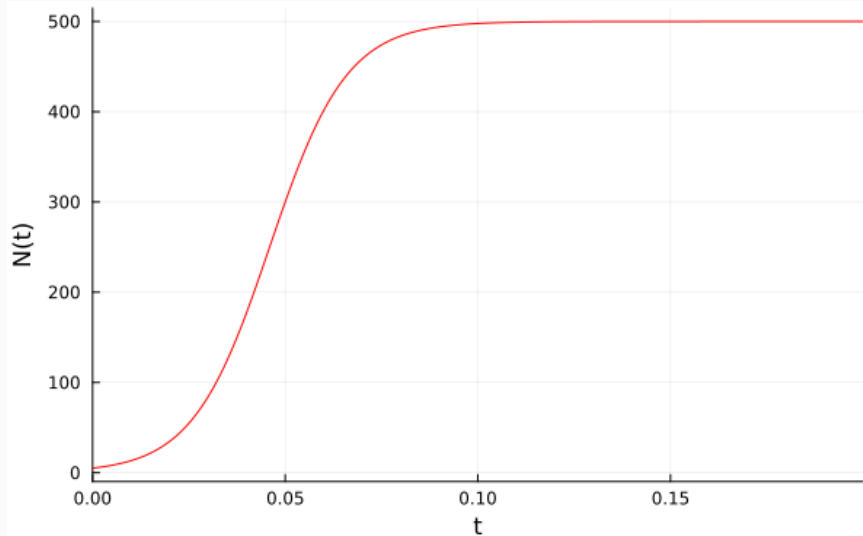
function F3(du, u, p, t)
    du[1] = (0.5*sin(t) + 0.3*cos(t)*u[1])*(N - u[1])
end

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)")

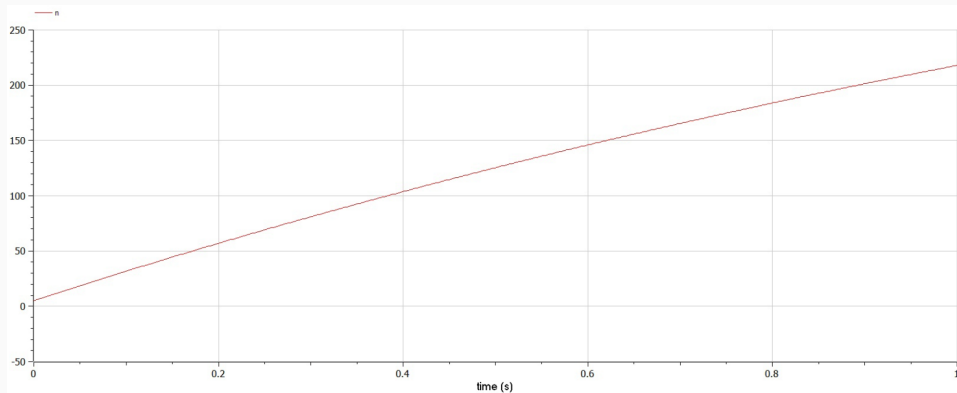
savefig(plt3, "lab7_3.png")
```

График распространения рекламы на языке Julia



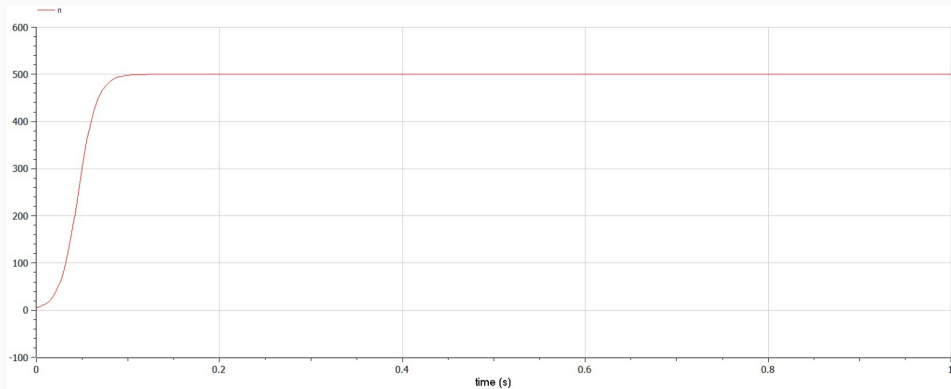
```
model Advert
parameter Real N = 500;
parameter Real N0 = 5;
Real n(start=N0);
equation
// 1 случай
der(n) = (0.55 + 0.0001*n)*(N - n);
end Advert;
```


График распространения рекламы на языке OpenModelica



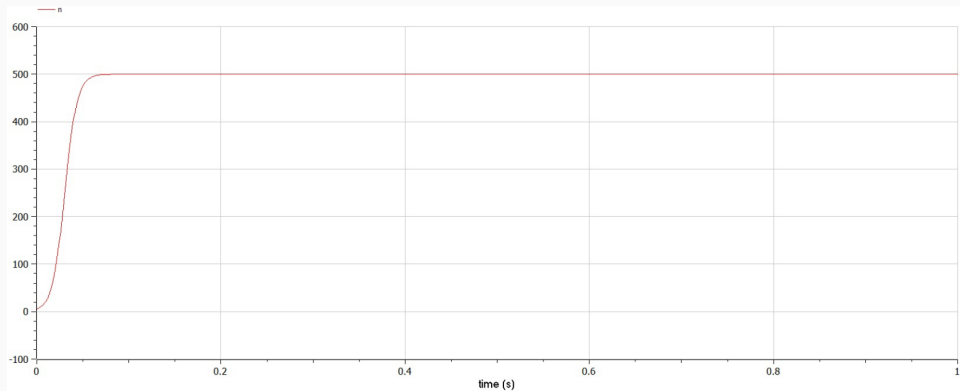
```
model Advert
parameter Real N = 500;
parameter Real N0 = 5;
Real n(start=N0);
equation
// 2 случай
der(n) = (0.00005 + 0.2*n)*(N - n);
end Advert;
```

График распространения рекламы на языке OpenModelica



```
model Advert
parameter Real N = 500;
parameter Real N0 = 5;
Real n(start=N0);
equation
// 3 случай
der(n) = (0.5*sin(time) + 0.3*cos(time)*n)*(N - n);
end Advert;
```

График распространения рекламы на языке OpenModelica



Результаты

- Улучшены навыки работы с Julia и OpenModelica
- Научились строить графики распространения рекламы
- Научились определять в какой момент времени скорость распространения рекламы будет иметь максимальное значение