

Лабораторная работа №7

Математическое моделирование

Соболевский Денис Андреевич

25 марта 2023

Российский университет дружбы народов, Москва, Россия

Информация

- Соболевский Денис Андреевич
- студент 3 курса группы НФИбд-02-20
- ст. б. 1032201668
- Российский университет дружбы народов

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

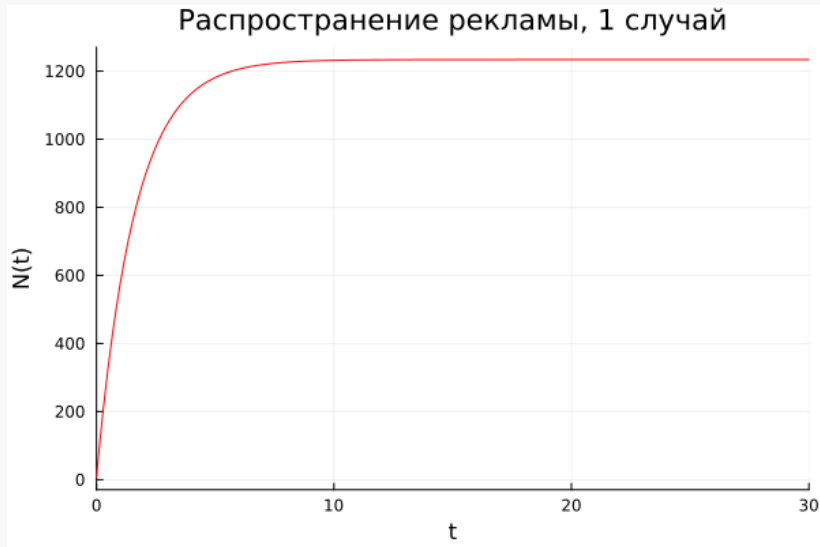
Модель рекламной кампании

- $\frac{dn}{dt} = (\alpha_1 + \alpha_2 n(t))(N - n(t))$
- $n(t)$ — 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

```
1 using Plots
2 using DifferentialEquations
3
4 const N = 1670
5 const N0 = 12
6
7 T1 = (0, 30)
8
9 T2 = (0, 0.25)
10
11 u0 = [N0]
12
13 # 1 случай (alpha1 >> alpha2)
14
15 function F1(du, u, p, t)
16     du[1] = (0.133 + 0.000033*u[1])*(N - u[1])
17 end
18
19 probl = ODEProblem(F1, u0, T1)
20 soll = solve(probl, dtmax=0.1)
21
22 plt1 = plot(soll, color=:red, title="Распространение рекламы, 1 случай", legend=false, xlabel="t", ylabel="N(t)")
23
24 savefig(plt1, "lab7_1.png")
25
```

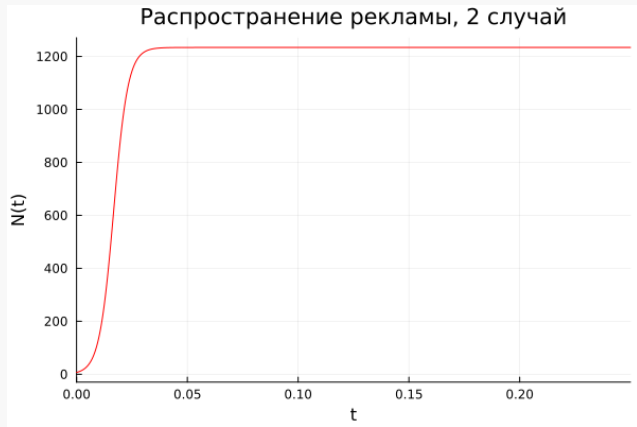
График распространения рекламы на языке Julia



Программа на языке Julia для второго случая

```
26 # 2 случай (alpha1 << alpha2)
27
28 maxx = [-10000.0, 0]
29
30 function F2(du, u, p, t)
31     du[1] = (0.0000132 + 0.32*u[1])*(N - u[1])
32
33     if du[1] > maxx[1]
34         maxx[1] = du[1]
35         maxx[2] = t
36     end
37 end
38
39 prob2 = ODEProblem(F2, u0, T2)
40 sol2 = solve(prob2, dtmax=0.001)
41
42 println("t = ", maxx[2])
43
44 plt2 = plot(sol2, color=:red, title="Распространение рекламы, 2 случай", legend=false, xlabel="t", ylabel="N(t)")
45
46 savefig(plt2, "lab7_2.png")
47
```

График распространения рекламы на языке Julia

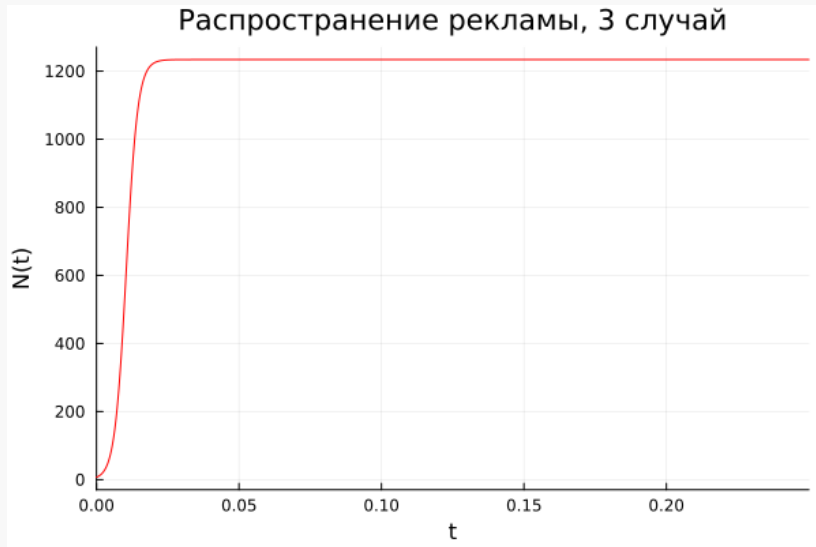


```
t = 0.016900000000000001
```

Программа на языке Julia для третьего случая

```
48 # 3 случай (alpha1, alpha2 - периодические функции)
49
50 function F3(du, u, p, t)
51     du[1] = (0.8(t) + 15*sin(t)*u[1])*(N - u[1])
52 end
53
54 prob3 = ODEProblem(F3, u0, T2)
55 sol3 = solve(prob3, dtmax=0.001)
56
57 plt3 = plot(sol3, color=:red, title="Распространение рекламы, 3 случай", legend=false, xlabel="t", ylabel="N(t)")
58
59 savefig(plt3, "lab7_3.png")
```

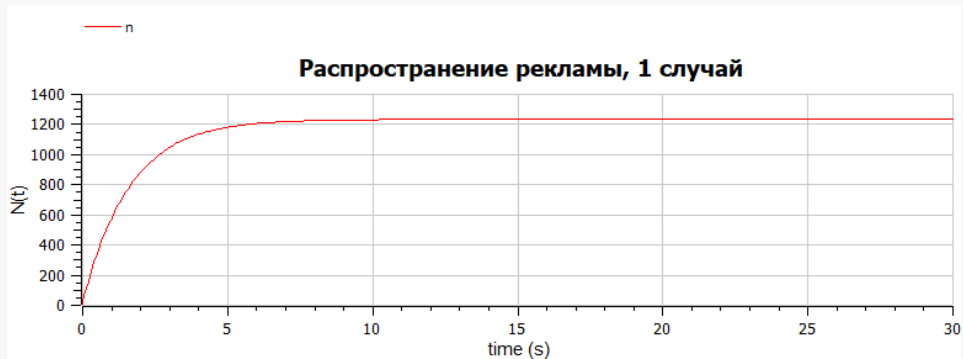
График распространения рекламы на языке Julia



Программа на языке OpenModelica для первого случая

```
1  model Advert
2  parameter Real N = 1670;
3  parameter Real N0 = 12;
4  Real n(start=N0);
5  equation
6  // 1 случай
7  der(n) = (0.133 + 0.000033*n)*(N - n);
8
9  end Advert;
10
```

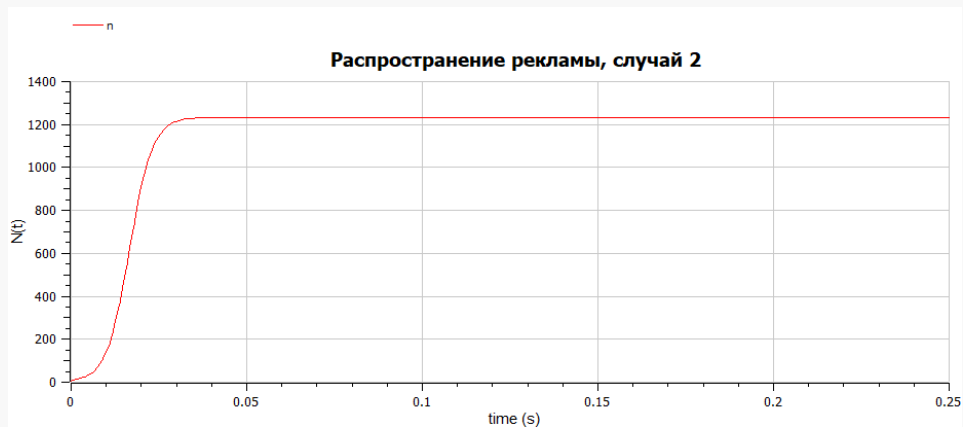

График распространения рекламы на языке OpenModelica



Программа на языке OpenModelica для второго случая

```
1 model Advert
2   parameter Real N = 1670;
3   parameter Real N0 = 12;
4   Real n(start=N0);
5   equation
6     // 2 случай
7     der(n) = (0.0000132 + 0.32*n) * (N - n);
8   end Advert;
9
```

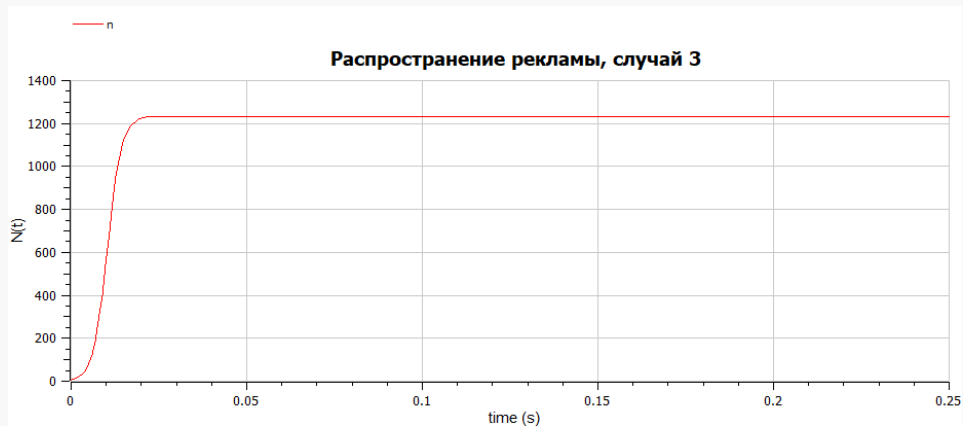
График распространения рекламы на языке OpenModelica



Программа на языке OpenModelica для третьего случая

```
1 model Advert
2   parameter Real N = 1670;
3   parameter Real N0 = 12;
4   Real n(start=N0);
5   equation
6     // 3 случай
7     der(n) = (0.8t+0.15sin*n) (N - n);
8   end Advert;
9
```

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