

Лабораторная работа №7

Эффективность рекламы

Кузнецов Юрий Владимирович

Содержание

Цель работы	1
Задачи.....	1
Среда	1
Теоретическое введение	2
Ход работы.....	2
Вывод.....	14
Ресурсы.....	14

Цель работы

Рассмотреть модель распространения информации о товаре (модель распространения рекламы). Построить вышеуказанную модель средствами OpenModelica и Julia.

Задачи

Построить график распространения рекламы, математическая модель которой описывается следующим уравнением

1. $\frac{dn}{dt} = (0.133 + 0.000033n(t))(N - n(t))$
2. $\frac{dn}{dt} = (0.0000132 + 0.32n(t))(N - n(t))$
3. $\frac{dn}{dt} = (0.8t + 0.15\sin(t)n(t))(N - n(t))$

При этом объем аудитории $N = 1670$, в начальный момент о товаре знает 12 человек. Для случая 2 определить в какой момент времени скорость распространения рекламы будет иметь максимальное значение.

Среда

- Julia – это открытый свободный высокопроизводительный динамический язык высокого уровня, созданный специально для технических

(математических) вычислений. Его синтаксис близок к синтаксису других сред технических вычислений, таких как Matlab и Octave. [@unn-julia]

- OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [@wiki-om]

Теоретическое введение


Предположим, что реализуется некоторая продукция, о которой в момент времени t из числа потенциальных покупателей N знает лишь n покупателей. Для ускорения сбыта продукции запускается реклама через СМИ. После запуска рекламной кампании информация о продукции начнет распространяться путем общения друг с другом. Таким образом, после запуска рекламных объявлений скорость изменения числа знающих о продукции людей пропорциональна как числу знающих о товаре, так и числу покупателей о нем не знающих.

Модель рекламной кампании описывается следующими величинами. Считаем, что $\frac{dn}{dt}$ - скорость изменения со временем числа потребителей, узнавших о товаре и готовых его купить, t - время, прошедшее с начала рекламной кампании, $n(t)$ - число уже информированных клиентов. Эта величина пропорциональна числу покупателей, еще не знающих о нем, это описывается следующим образом: $\alpha_1(t)(N - n(t))$, где N - общее число потенциальных покупателей, $\alpha_1(t) > 0$ - характеризует интенсивность рекламной кампании (зависит от затрат на рекламу в данный момент времени). Помимо этого, узнавшие о товаре потребители также распространяют полученную информацию среди не знающих о нем. Этот вклад в рекламу описывается величиной $\alpha_2(t)n(t)(N - n(t))$, эта величина увеличивается с увеличением потребителей, узнавших о товаре. Математическая модель распространения рекламы описывается уравнением:

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

Ход работы

Запускаем Pluto.

Save notebook...

```
using Plots ✓  
  
using DifferentialEquations ✓  
  
[12]  
begin  
    const N = 1670  
    u0 = [12]  
end
```

Julia. Скриншот (1). Эффективность рекламы

Построим модель при помощи DifferentialEquations и решим систему ДУ. Построим график решения и сохраним его. Запустим код.

```
alpha1 = 0.133  
alpha2 = 0.000033  
t = (0, 30)  
  
function AD!(du, u, p, t)  
    du[1] = (alpha1 + alpha2 * u[1]) * (N - u[1])  
end  
  
prob = ODEProblem(AD!, u0, t)  
sol = solve(prob)  
  
plt = plot(  
    sol,  
    dpi=500,  
    size=(1024, 512),  
    plot_title="Эффективность рекламы",  
    xlabel="Время",  
    ylabel="n(t)",  
    label="n(t) - количество заинтересованных в товаре людей")  
  
savefig(plt, "labart/result.png")  
println("Complete!")
```

```
• using Plots ✓
```

```
• using DifferentialEquations ✓
```

```
► [12]
```

```
• begin
•     const N = 1670
•     u0 = [12]
• end

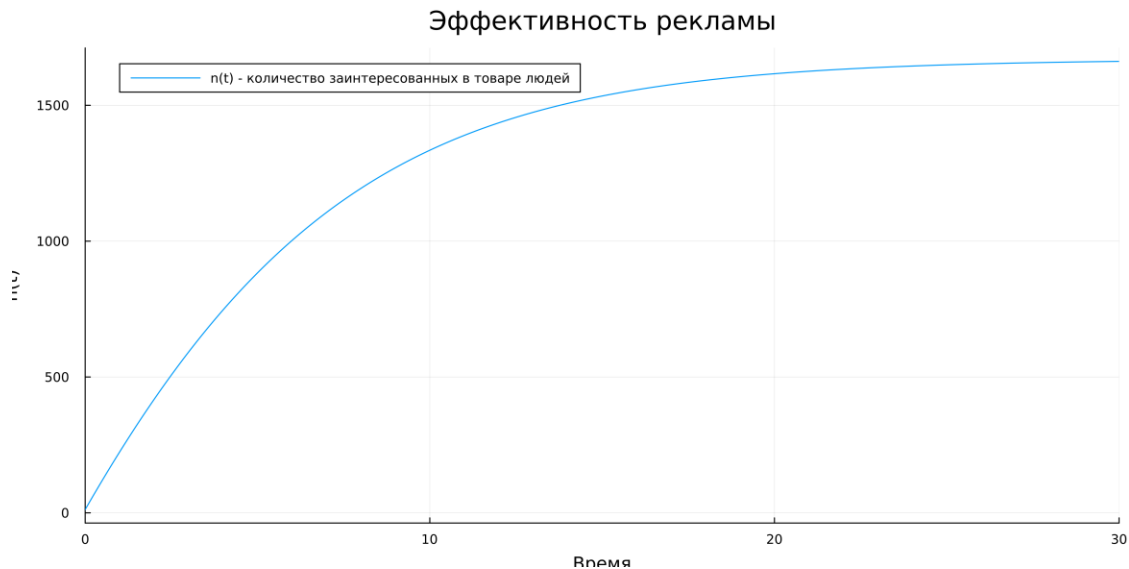
• begin
•     alpha1 = 0.133
•     alpha2 = 0.000033
•     t = (0, 30)
•
•     function AD!(du, u, p, t)
•         du[1] = (alpha1 + alpha2 * u[1]) * (N - u[1])
•     end
•
•     prob = ODEProblem(AD!, u0, t)
•     sol = solve(prob)
•
•     plt = plot(
•         sol,
•         dpi=500,
•         size=(1024, 512),
•         plot_title="Эффективность рекламы",
•         xlabel="Время",
•         ylabel="n(t)",
•         label="n(t) - количество заинтересованных в товаре людей")
•
•     savefig(plt, "labart/result.png")
•     println("Complete!")
• end
```



```
Complete!
```



Julia. Скрипт (2). Эффективность рекламы ($\alpha_1 = 0.133$, $\alpha_2 = 0.000033$)



Julia. Модель. Эффективность рекламы ($\alpha_1 = 0.133$, $\alpha_2 = 0.000033$)

Изменим значения коэффициентов α , так же модернизируем функцию системы, чтобы найти максимальное значение и будем сохранять его в заранее заданную переменную. После чего на графике решения системы отобразим точку, которая соответствует максимальной скорости распространения рекламы.

```
alpha1 = 0.0000132
alpha2 = 0.32
t = (0, 0.02)
max_speed = [-1e12, 0, 0]

function AD!(du, u, p, t)
    du[1] = (alpha1 + alpha2 * u[1]) * (N - u[1])
    if du[1] > max_speed[1]
        max_speed[1] = du[1]
        max_speed[2] = t
        max_speed[3] = u[1]
    end
end

end

prob = ODEProblem(AD!, u0, t)
sol = solve(prob)

plt = plot(
    sol,
    dpi=500,
    size=(1024, 512),
    plot_title="Эффективность рекламы",
    xlabel="Время",
    ylabel="n(t)",
    label="количество заинтересованных в товаре людей")
```

```

scatter!(
    plt,
    [max_speed[2]],
    [max_speed[3]],
    seriestype=:scatter,
    label="максимальное значение скорости распространения рекламы")

savefig(plt, "labart/result2.png")
println(max_speed)
println("OK!")

```

```

begin
    alpha1 = 0.0000132
    alpha2 = 0.32
    t = (0, 0.02)
    max_speed = [-1e12, 0, 0]

    function AD!(du, u, p, t)
        du[1] = (alpha1 + alpha2 * u[1]) * (N - u[1])
        if du[1] > max_speed[1]
            max_speed[1] = du[1]
            max_speed[2] = t
            max_speed[3] = u[1]
        end
    end

    prob = ODEProblem(AD!, u0, t)
    sol = solve(prob)

    plt = plot(
        sol,
        dpi=500,
        size=(1024, 512),
        plot_title="Эффективность рекламы",
        xlabel="Время",
        ylabel="n(t)",
        label="количество заинтересованных в товаре людей")

    scatter!(
        plt,
        [max_speed[2]],
        [max_speed[3]],
        seriestype=:scatter,
        label="максимальное значение скорости распространения рекламы")

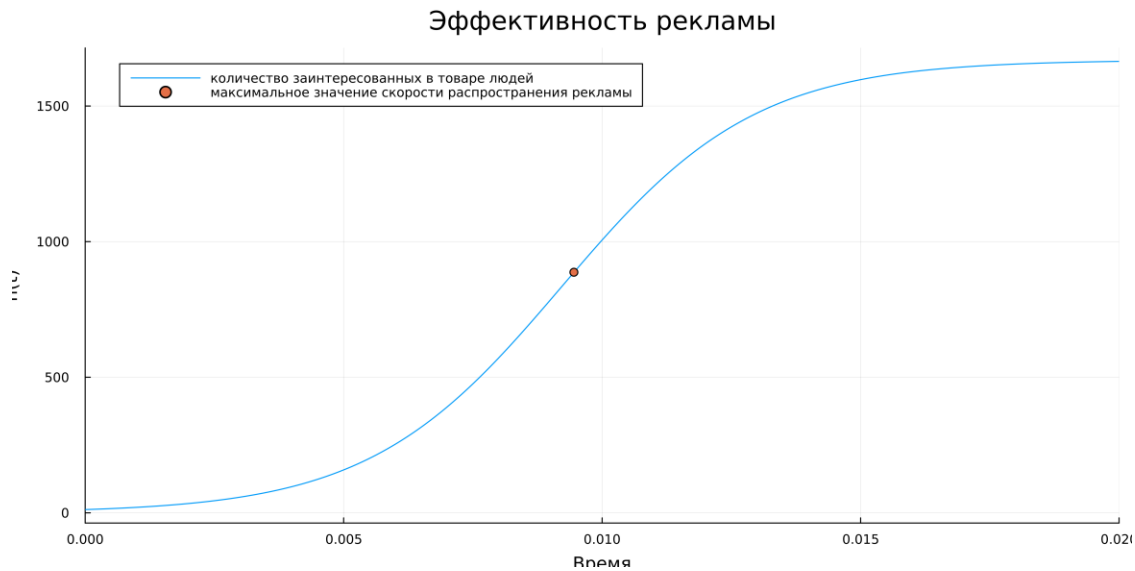
    savefig(plt, "labart/result2.png")
    println(max_speed)
    println("OK!")

end

```

[222241.60624150743, 0.009453997858304789, 887.1537418379303]
?OK!

Julia. Скрипт (3). Эффективность рекламы ($\alpha_1 = 0.0000132$, $\alpha_2 = 0.32$)



Julia. Модель. Эффективность рекламы ($\alpha_1 = 0.0000132$, $\alpha_2 = 0.32$)

Изменим скрипт 1го кода, а именно поменяем α , промежуток времени, а также изменим функцию в соответствии с задачей.

```
alpha1 = 0.8
alpha2 = 0.15
t = (0, 0.4)
```

```
function AD!(du, u, p, t)
    du[1] = (alpha1 * t + alpha2 * sin(t) * u[1]) * (N - u[1])
end
```

```
prob = ODEProblem(AD!, u0, t)
sol = solve(prob)
```

```
plt = plot(
    sol,
    dpi=500,
    size=(1024, 512),
    plot_title="Эффективность рекламы",
    xlabel="Время",
    ylabel="n(t)",
    label="n(t) - количество заинтересованных в товаре людей")
```

```
savefig(plt, "labart/result3.png")
println("Success!")
```



```

begin
  alpha1 = 0.8
  alpha2 = 0.15
  t = (0, 0.4)

  function AD!(du, u, p, t)
    du[1] = (alpha1 * t + alpha2 * sin(t) * u[1]) * (N - u[1])
  end

  prob = ODEProblem(AD!, u0, t)
  sol = solve(prob)

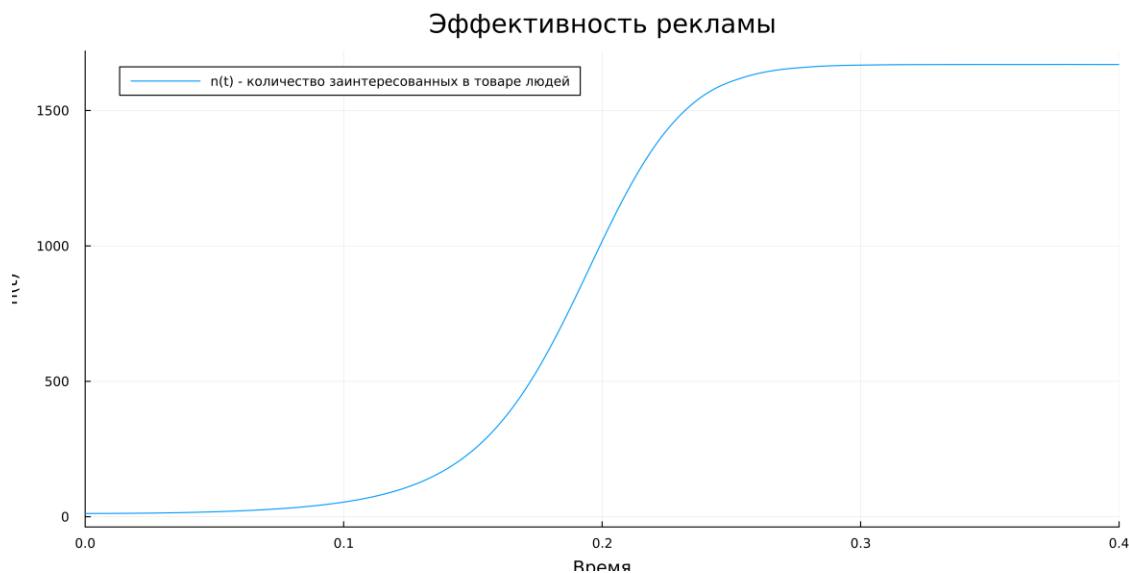
  plt = plot(
    sol,
    dpi=500,
    size=(1024, 512),
    plot_title="Эффективность рекламы",
    xlabel="Время",
    ylabel="n(t)",
    label="n(t) - количество заинтересованных в товаре людей")

  savefig(plt, "labart/result3.png")
  println("Success!")
end

```

Success!

Julia. Скрипт (4). Эффективность рекламы ($\alpha_1 = 0.8, \alpha_2 = 0.15$)



Julia. Модель. Эффективность рекламы ($\alpha_1 = 0.8, \alpha_2 = 0.15$)

Напишем код на modelica для решения 1-ой задачи. После чего запустим его и сохраним график.

```

model Var1
  constant Integer N = 1670;
  constant Real alpha1 = 0.133;

```

```

    constant Real alpha2 = 0.000033;
    Real t = time;
    Real n(t);
initial equation
    n = 12;
equation
    der(n) = (alpha1 + alpha2 * n) * (N - n);
    annotation(experiment(StartTime = 0, StopTime = 30, Interval = 0.001));
end Var1;

```

```

1  model Var1
2      constant Integer N = 1670;
3      constant Real alpha1 = 0.133;
4      constant Real alpha2 = 0.000033;
5      Real t = time;
6      Real n(t);
7  initial equation
8      n = 12;
9  equation
10     der(n) = (alpha1 + alpha2 * n) * (N - n);
11     annotation(experiment(StartTime = 0, StopTime = 30, Interval = 0.001));
12 end Var1;
13
14
15
16
17
18
19

```

Ln: 16, Col: 0

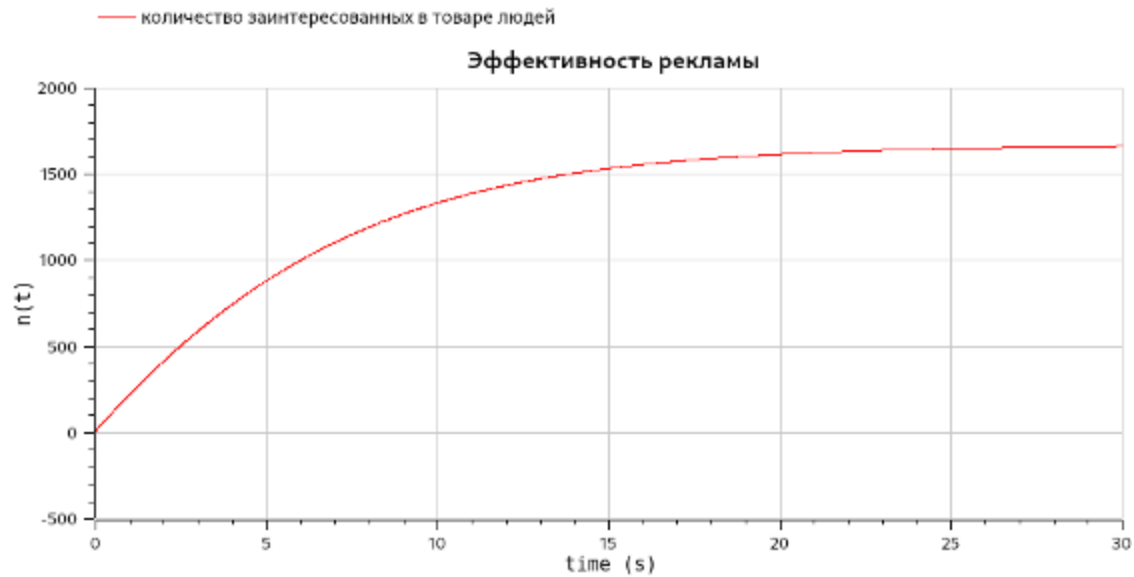
Welcome

Modeling

Plotting

Debugging

Modelica. Скринт. Эффективность рекламы ($\alpha_1 = 0.133$, $\alpha_2 = 0.000033$)



Modelica. Модель. Эффективность рекламы ($\alpha_1 = 0.133$, $\alpha_2 = 0.000033$)

Напишем скрипт на modelica для решения 2-ой задачи: изменим начальные значения. После чего запустим его и сохраним график.

```
model Var2
  constant Integer N = 1670;
  constant Real alpha1 = 0.0000132;
  constant Real alpha2 = 0.32;
  Real t = time;
  Real n(t);
initial equation
  n = 12;
equation
  der(n) = (alpha1 + alpha2 * n) * (N - n);
  annotation(experiment(StartTime = 0, StopTime = 0.02, Interval = 0.001));
end Var2;
```

```

1 model Var2
2   constant Integer N = 1670;
3   constant Real alpha1 = 0.0000132;
4   constant Real alpha2 = 0.32;
5   Real t = time;
6   Real n(t);
7   initial equation
8     n = 12;
9   equation
10    der(n) = (alpha1 + alpha2 * n) * (N - n);
11    annotation(experiment(StartTime = 0, StopTime = 0.02, Interval = 0.001));
12  end Var2;
13
14
15
16
17
18
19
20

```

Ln: 19, Col: 0

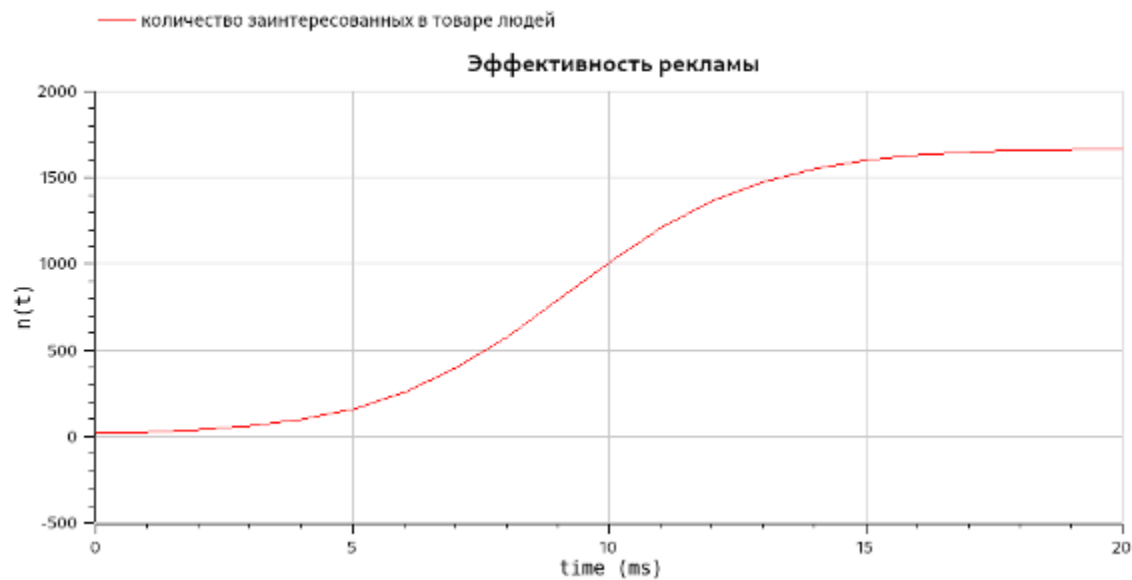
Welcome

Modeling

Plotting

Debugging

Modelica. Скринш. Эффективность рекламы ($\alpha_1 = 0.0000132$, $\alpha_2 = 0.32$)



Modelica. Модель. Эффективность рекламы ($\alpha_1 = 0.0000132$, $\alpha_2 = 0.32$)

Напишем скрипт на modelica для решения 3-ей задачи: изменим начальные значения, а также уравнение. После чего запустим его и сохраним график.

```

model Var3
  constant Integer N = 1670;
  constant Real alpha1 = 0.8;

```

```





    constant Real alpha2 = 0.15;
    Real t = time;
    Real n(t);
initial equation
    n = 12;
equation
    der(n) = (alpha1 * t + alpha2 * sin(t) * n) * (N - n);
    annotation(experiment(StartTime = 0, StopTime = 0.4, Interval = 0.001));
end Var3;

```

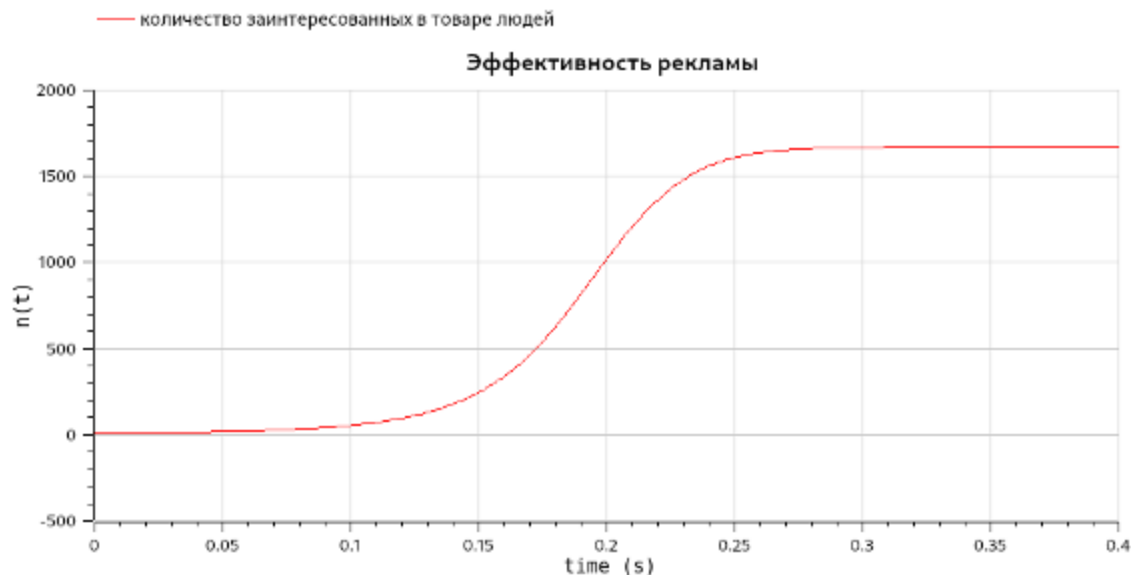
```

1  model Var3
2      constant Integer N = 1670;
3      constant Real alpha1 = 0.8;
4      constant Real alpha2 = 0.15;
5      Real t = time;
6      Real n(t);
7  initial equation
8      n = 12;
9  equation
10     der(n) = (alpha1 * t + alpha2 * sin(t) * n) * (N - n);
11     annotation(experiment(StartTime = 0, StopTime = 0.4, Interval = 0.001));
12 end Var3;
13
14
15
16
17
18
19
20
21

```

Ln: 19, Col: 0  Welcome  Modeling  Plotting  Debugging

Modelica. Скрипт. Эффективность рекламы ($\alpha_1 = 0.8$, $\alpha_2 = 0.15$)



Modelica. Модель. Эффективность рекламы ($\alpha_1 = 0.8$, $\alpha_2 = 0.15$)

Вывод

Повысили навыки в решении ДУ, поближе познакомились с моделированием на Julia и OpenModelica. Изучили и построили модель распространения рекламы.

Ресурсы

- Julia. URL: http://www.unn.ru/books/met_files/JULIA_tutorial.pdf.
- OpenModelica [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.
- Эффективность рекламы. RUDN. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=967253>.
- Pluto. URL: <https://plutojl.org/>.
- Plots in Julia. URL: <https://docs.juliaplots.org/latest/tutorial/>.
- Differential Equations in Julia. URL: https://docs.sciml.ai/DiffEqDocs/stable/getting_started/.