Лабораторная работа №7

Эффективность рекламы

Ильин Андрей Владимирович

Содержание

# 1 Цель работы

Рассмотреть модель распространениинформации о товаре (модель распространения рекламы). Построить вышеуказанную модель средствами OpenModellica и Julia.

# 2 Задачи

Построить график распространения рекламы, математическая модель которой описывается следующим уравнением

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

# 3 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. [1]
* OpenModelica — свободное открытое программное обеспечение для моделирования, симуляции, оптимизации и анализа сложных динамических систем. Основано на языке Modelica. [2]

# 4 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. [3]

Модель рекламной кампании описывается следующими величинами. Считаем, что - 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: ,где N - общее число потенциальных покупателей, - 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 , эта величина увеличивается с увеличением потребителей узнавших о товаре. Математическая модель распространения рекламы описывается уравнением:

# 5 Выполнение лабораторной работы

1. Начнем выполнения поставленных задач в Julia. Для этого запустим Pluto [4]. (рис. [1](#fig:001))

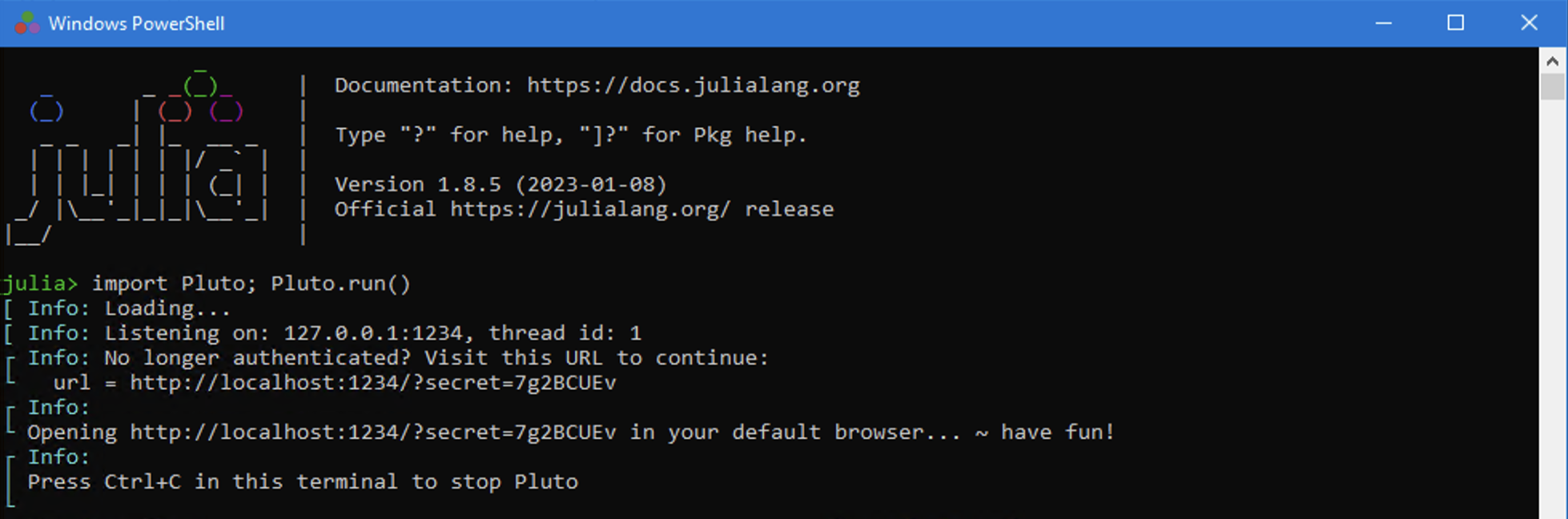


Figure 1: Julia. Запуск Pluto

1. Первым делом подкючим пакеты “Plots” [5] и “DifferentialEquations” [6]. Далее объявим начальные данные верные для всех кейсов при помощи констант. Также объявим начальное условие для системы ДУ. (рис. [2](#fig:002))

# подключение пакетов  
using Plots  
using DifferentialEquations  
  
# входные данные  
const N = 1670  
u0 = [12]

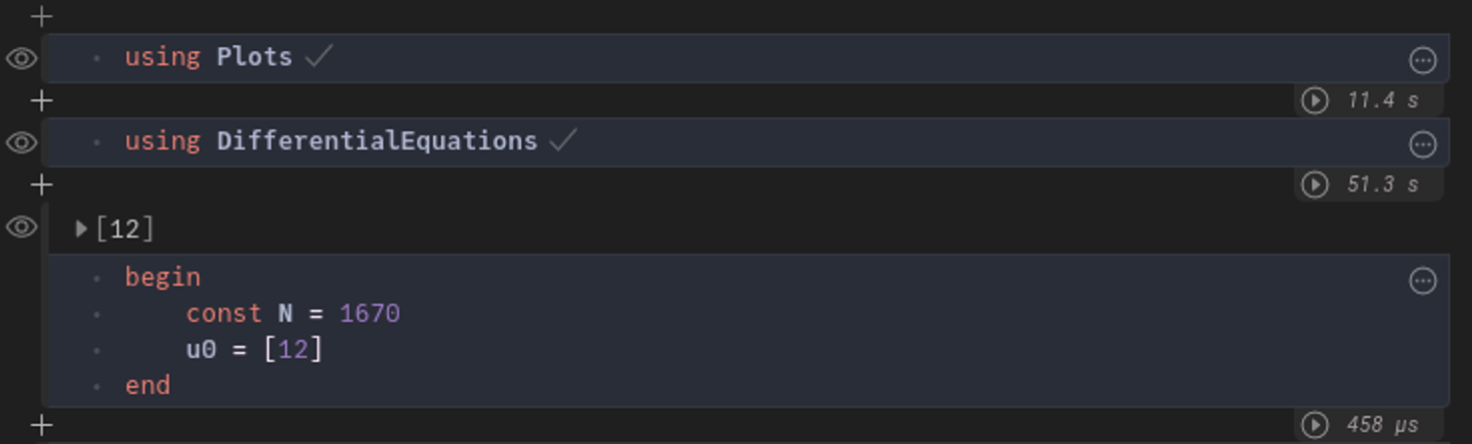


Figure 2: Julia. Скрипт (1). Эффективность рекламы

1. В следующей ячейке Pluto построим модель. При помощи ‘DifferentialEquations’ зададим и решим систему ДУ, после чего построим график решения и сохраним его. Далее запустим итоговый скрипт. (рис. [3](#fig:003), [4](#fig:004))

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, "artifacts/JL.lab07-01.png")  
println("Success")



Figure 3: Julia. Скрипт (2). Эффективность рекламы (, )

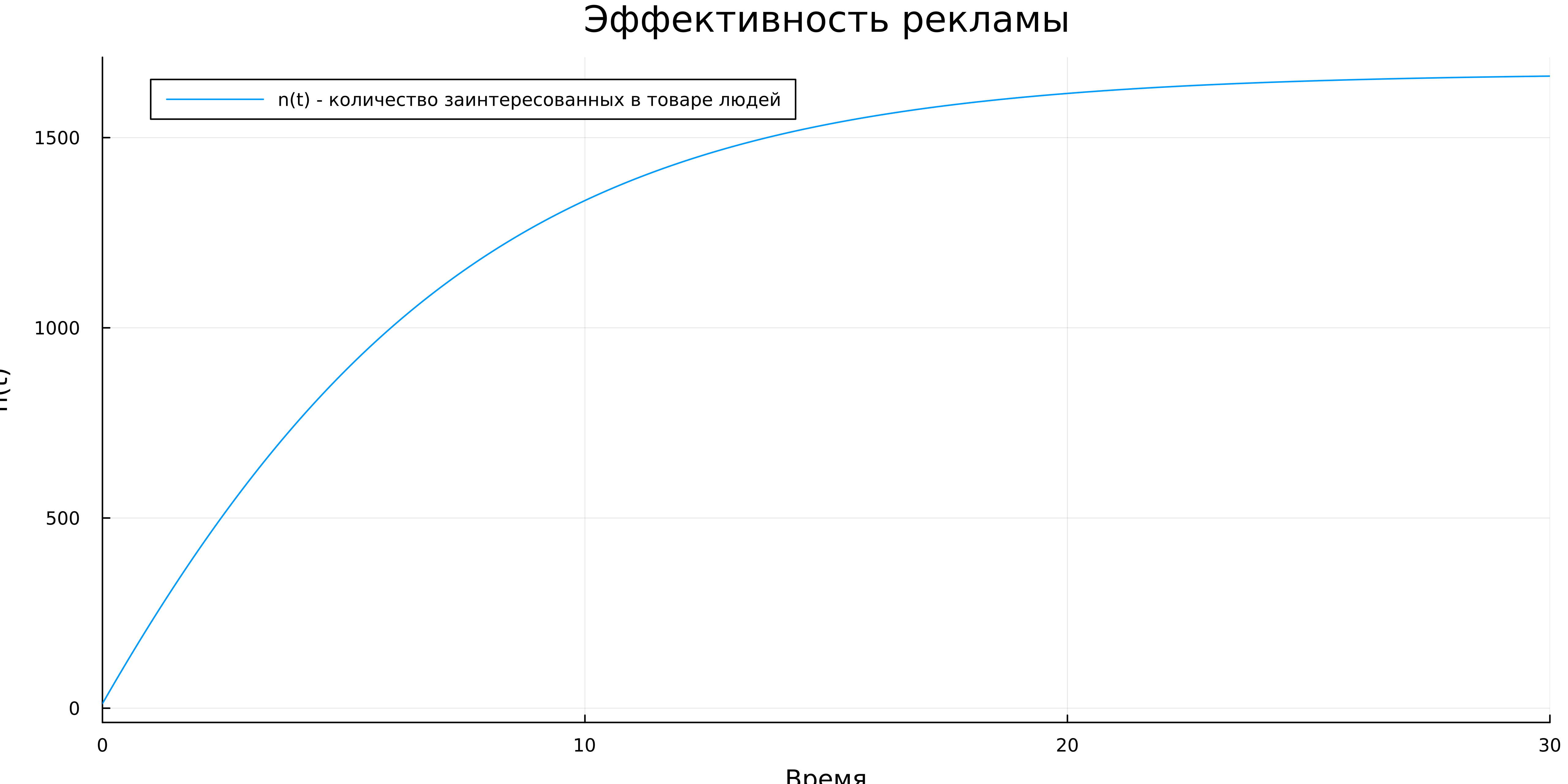


Figure 4: Julia. Модель. Эффективность рекламы (, )

1. Изменим значения коэффициентов , так же модернизируем функцию системы, чтобы найти максимальное значение и будем сохранять его в заранне заданную переменную. После чего на графике реения системы отобразим точку, которая соответствует максимальной скорости распространения рекламы. (рис. [5](#fig:005), [6](#fig:006))

alpha1 = 0.0000132 #!  
alpha2 = 0.32 #!  
t = (0, 0.02) #!  
max\_speed = [-1e12, 0, 0] #макс значение производной + (t и значение n(t))  
  
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 cкорости распространения рекламы")  
  
savefig(plt, "artifacts/JL.lab07-02.png")  
println(max\_speed)  
println("Success")



Figure 5: Julia. Скрипт (3). Эффективность рекламы (, )

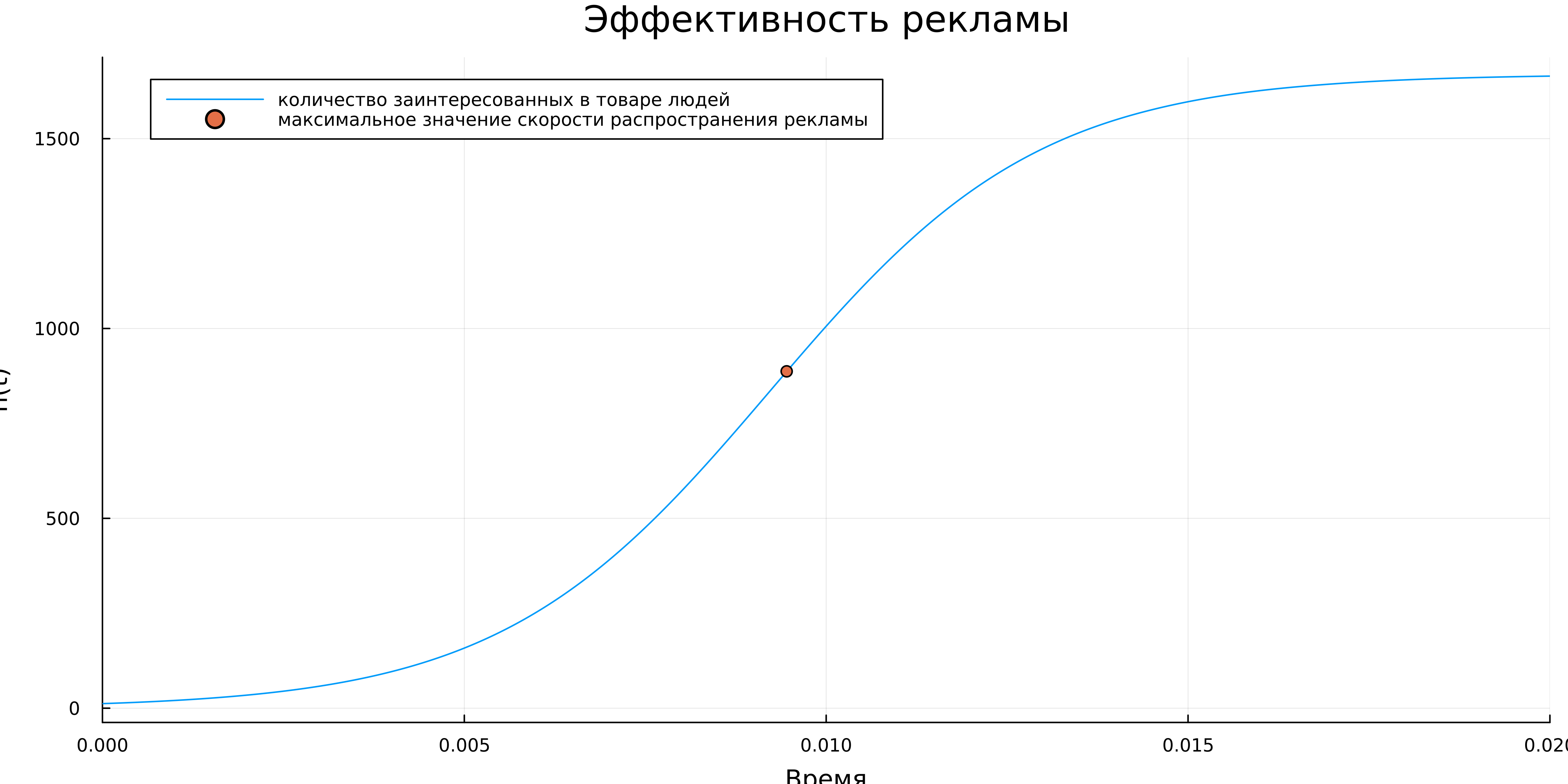


Figure 6: Julia. Модель. Эффективность рекламы (, )

1. Изменим скрипт 1го кейса, а именно поменяем , промежуток времени, а также изменим функцию в соотвествии с задачей. (рис. [7](#fig:007), [8](#fig:008))

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, "artifacts/JL.lab07-03.png")  
println("Success")

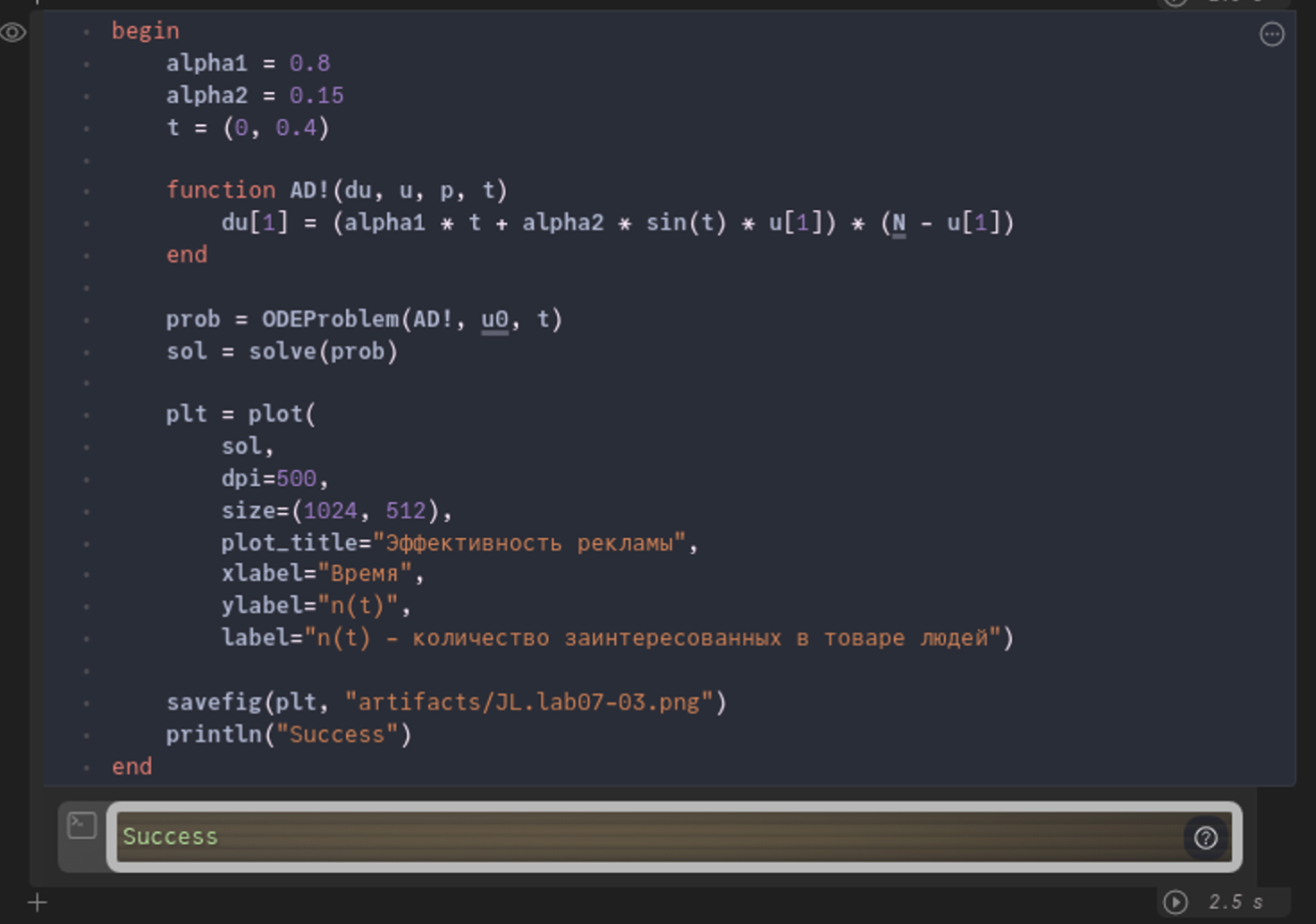


Figure 7: Julia. Скрипт (4). Эффективность рекламы (, )

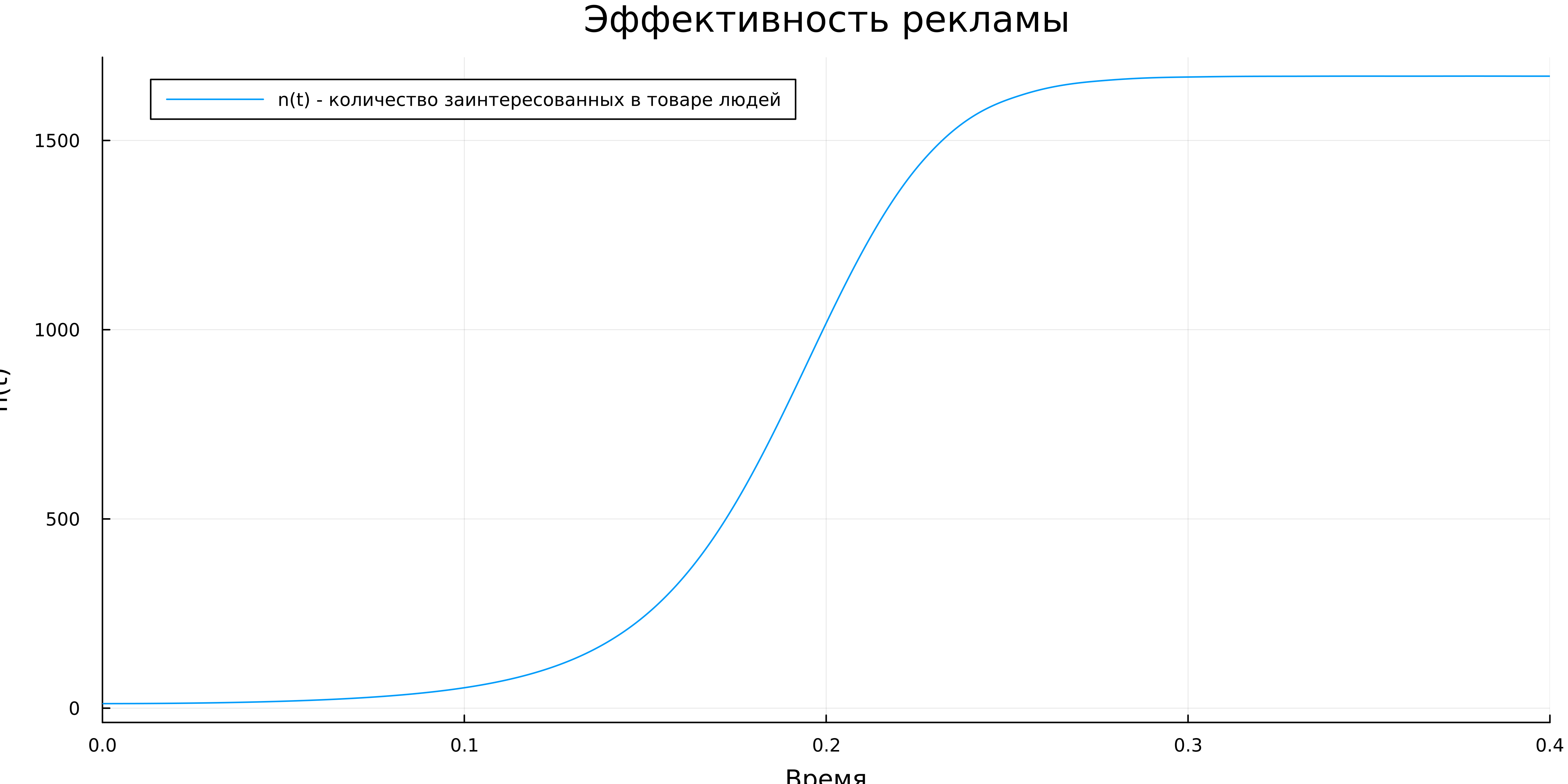


Figure 8: Julia. Модель. Эффективность рекламы (, )

1. Напишем скрипт на modellica для решения 1-ой задачи. После чего запустим его и сохраним график. (рис. [9](#fig:009), [10](#fig:010))

model lab07\_1  
 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 lab07\_1;

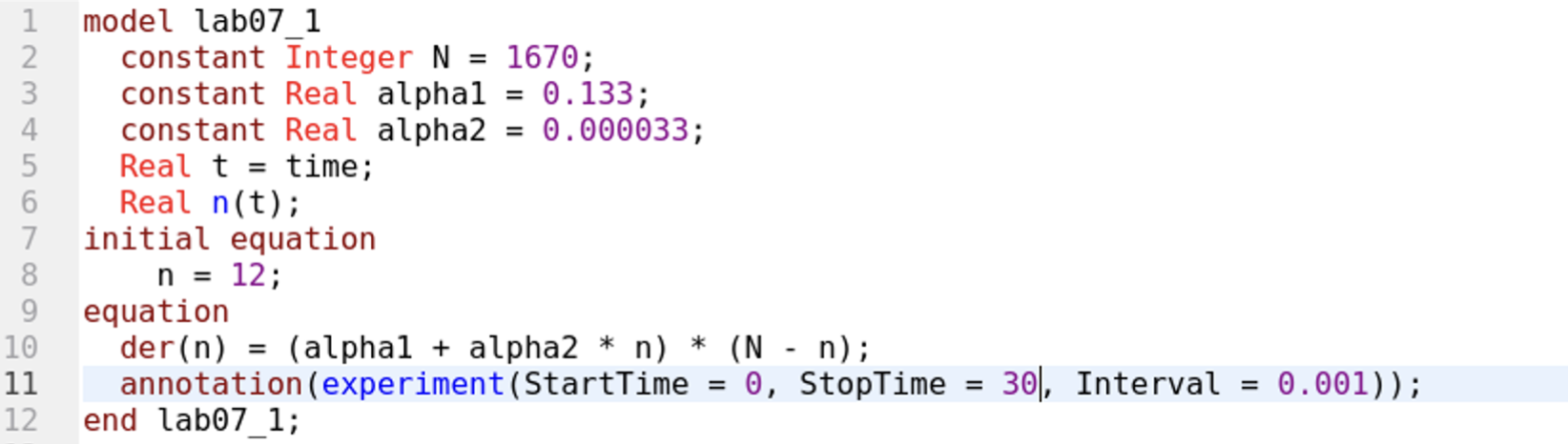


Figure 9: Modelica. Скрипт. Эффективность рекламы (, )

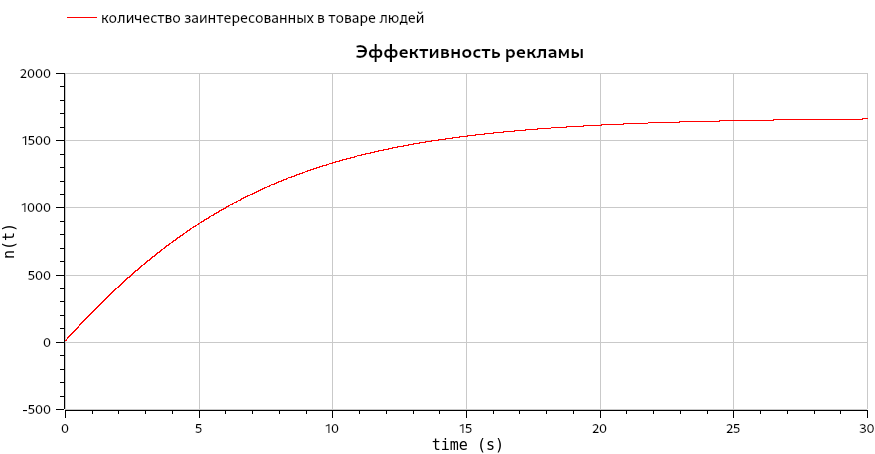


Figure 10: Modelica. Модель. Эффективность рекламы (, )

1. Напишем скрипт на modellica для решения 2-ой задачи: изменим начальные значения. После чего запустим его и сохраним график. (рис. [11](#fig:011), [12](#fig:012))

model lab07\_2  
 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 lab07\_2;

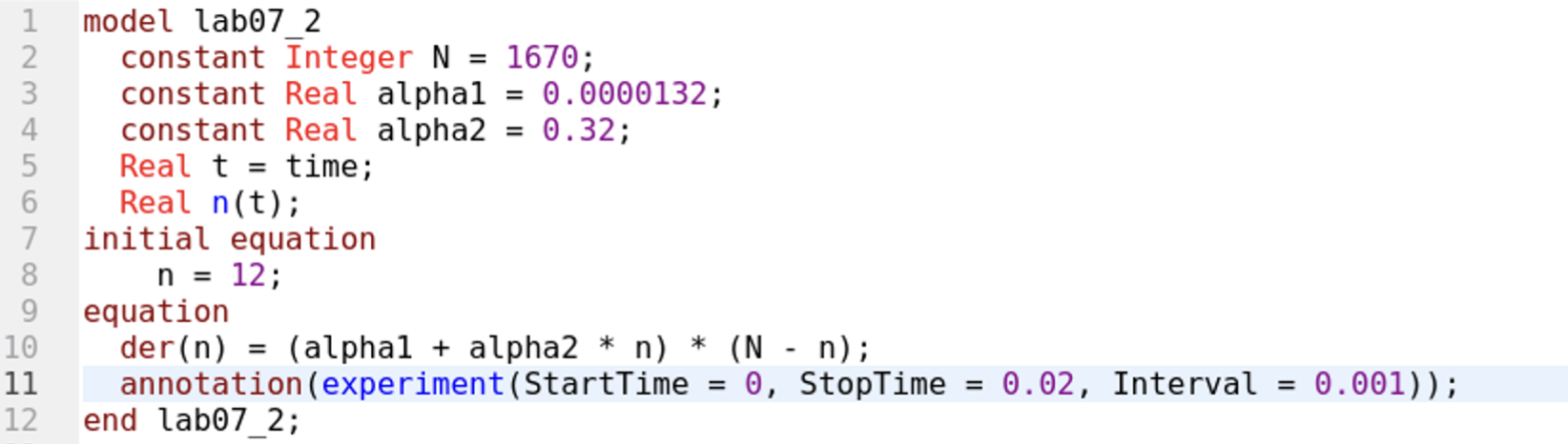


Figure 11: Modelica. Скрипт. Эффективность рекламы (, )

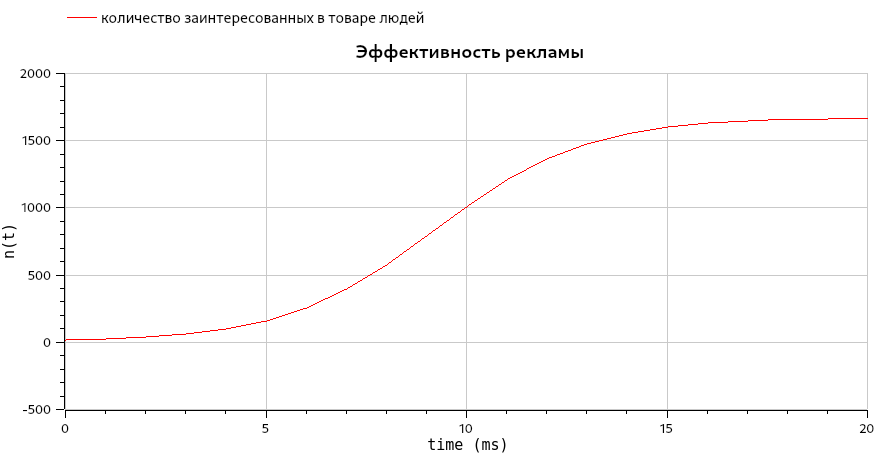


Figure 12: Modelica. Модель. Эффективность рекламы (, )

1. Напишем скрипт на modellica для решения 3-ой задачи: изменим начальные значения, а также уравнение. После чего запустим его и сохраним график. (рис. [13](#fig:013), [14](#fig:014))

model lab07\_3  
 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 lab07\_3;

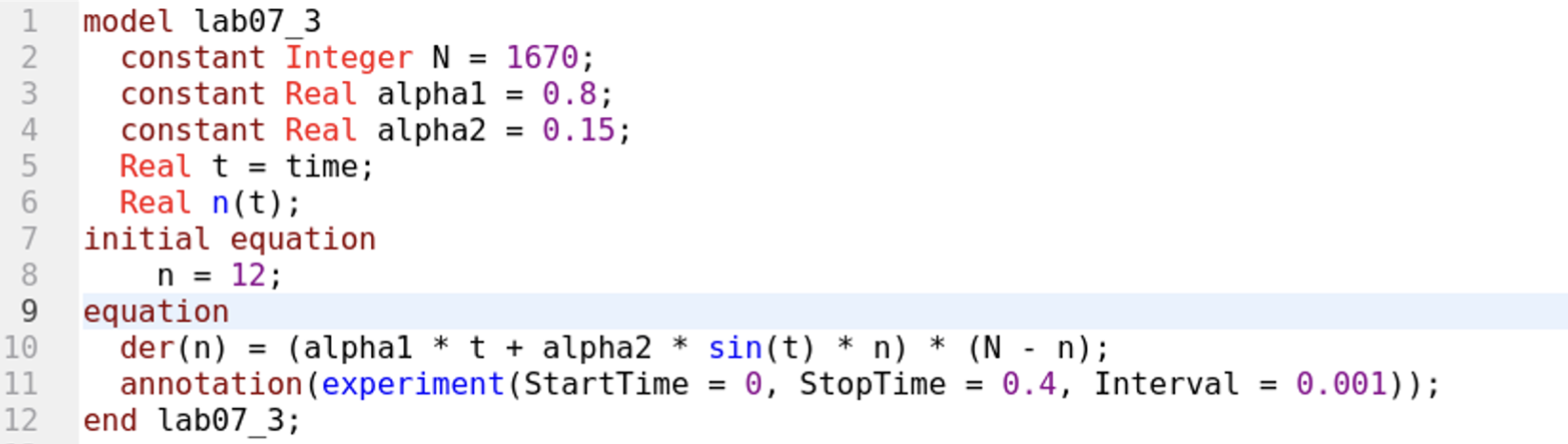


Figure 13: Modelica. Скрипт. Эффективность рекламы (, )

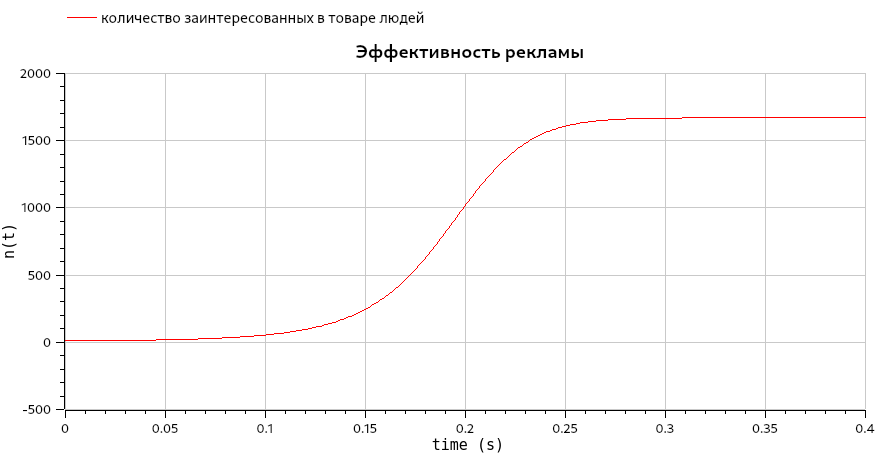


Figure 14: Modelica. Модель. Эффективность рекламы (, )

# 6 Анализ результатов

Работа выполненна без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

Моделирование на OMEdit было проще и быстрее, чем при использовании средств Julia. Скрипт на Modelica вышел более лакончиным, понятным и коротким. Более того OpenModelica быстрее обрабатывала скрипт и симмулировала модель. Стоит отметить, что OpenModelica имеет множество разлиных полезных инструментов для настройки с симмуляцией и работой с ней.

К плюсам Julia можно отнести, что она является языком программирования, который хорошо подходит для математических и технических задач. Отметим, что скрипт на Julia выполняется долго из-за подключения пакетов, каждый раз при его запуске. При использовании Pluto, нет необходимости каждый раз с нуля выполнять скрипт, таким образом скорость выполнения может даже превышать скорость моделирования в OMEdit.

# 7 Выводы

Мы улучшили практические навыки в области дифференциальных уравнений, улучшили навыки моделирования на Julia, а также навыки моделирования на OpenModelica. Изучили и построили модель распространения рекламы.

# Список литературы

1. Julia [Электронный ресурс]. URL: <http://www.unn.ru/books/met_files/JULIA_tutorial.pdf>.

2. OpenModelica [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/OpenModelica>.

3. Эффективность рекламы [Электронный ресурс]. RUDN. URL: <https://esystem.rudn.ru/mod/resource/view.php?id=967253>.

4. Pluto [Электронный ресурс]. URL: <https://plutojl.org/>.

5. Plots in Julia [Электронный ресурс]. URL: <https://docs.juliaplots.org/latest/tutorial/>.

6. Differential Equations in Julia [Электронный ресурс]. URL: <https://docs.sciml.ai/DiffEqDocs/stable/getting_started/>.