Лабораторная работа н.7

Задач эффективности рекламы

Петров Артем Евгеньевич

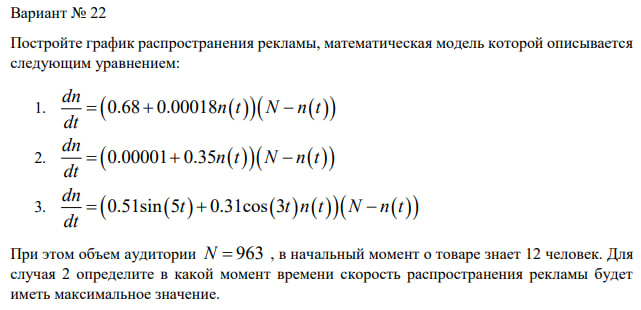
23 Марта 2024

# Информация

## Докладчик

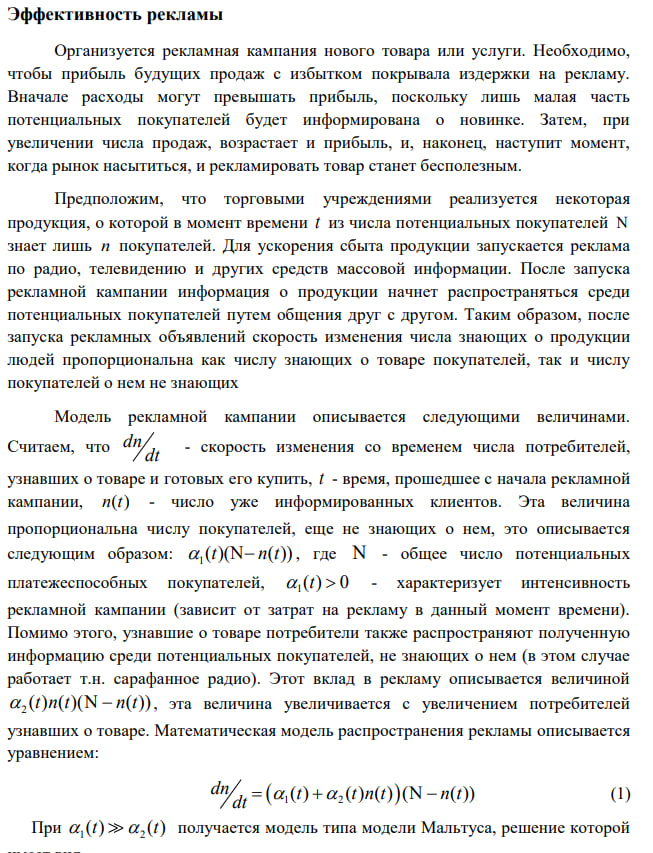
* Петров Артем Евгеньевич
* Студент
* Российский университет дружбы народов
* [1032219251@rudn.ru](mailto:1032219251@rudn.ru)
* <https://github.com/wlcmtunknwndth>

# Задание

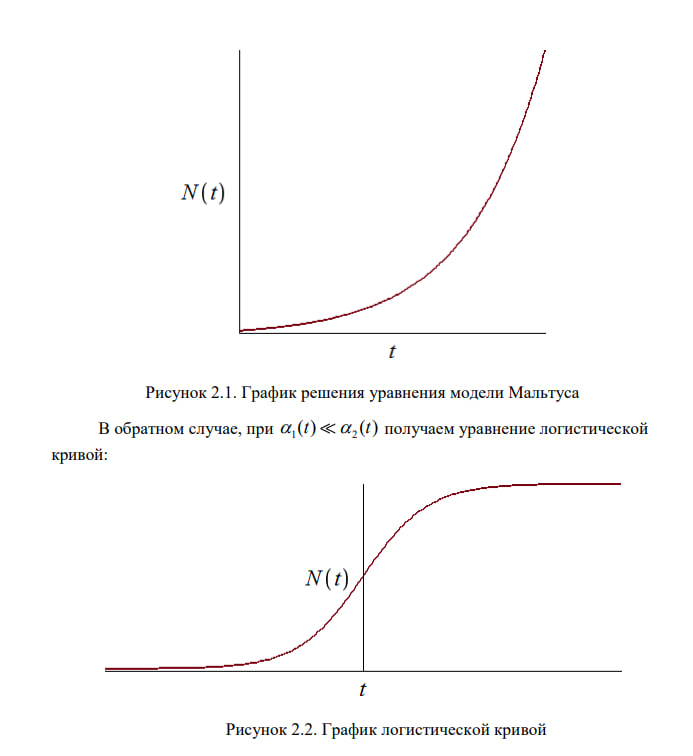


Задание

# Теоретическое введение(рис. 1)



Теоретическое введение ч.1



Теоретическое введение ч.2

# Выполнение лабораторной работы

## 1. Подключение необходимых библиотек

Подключим необходимые библиотеки:

using Plots  
using DifferentialEquations

## 2. Выполнение лабораторной для задачи 1.

### Код программы:

N = 963  
n0 = 12  
  
  
alpha = 0.68  
beta = 0.00018  
  
function ode(du, u, p, t)  
 du[1] = (alpha + beta\*u[1]) \* (N - u[1])  
   
end  
  
t\_arr = (0, 30)  
problem = ODEProblem(ode, [n0], t\_arr)  
  
solution = solve(problem, dtmax = 0.05)  
  
n = [u[1] for u in solution.u]  
T = [t for t in solution.t]  
  
plt = plot(  
 dpi = 500,   
 title = "Advertisement efficiency",  
 xlabel = "time",  
 ylabel = "people know",  
 legend = true  
)  
  
plot!(  
 plt,  
 T,  
 n,  
 label = "efficiency",  
 color = :green  
)  
  
savefig(plt, "./lab7/image/lab7\_1.png")

### График эффективности рекламы для задачи 1.

В итоге, получим вот такой график(рис. 1):

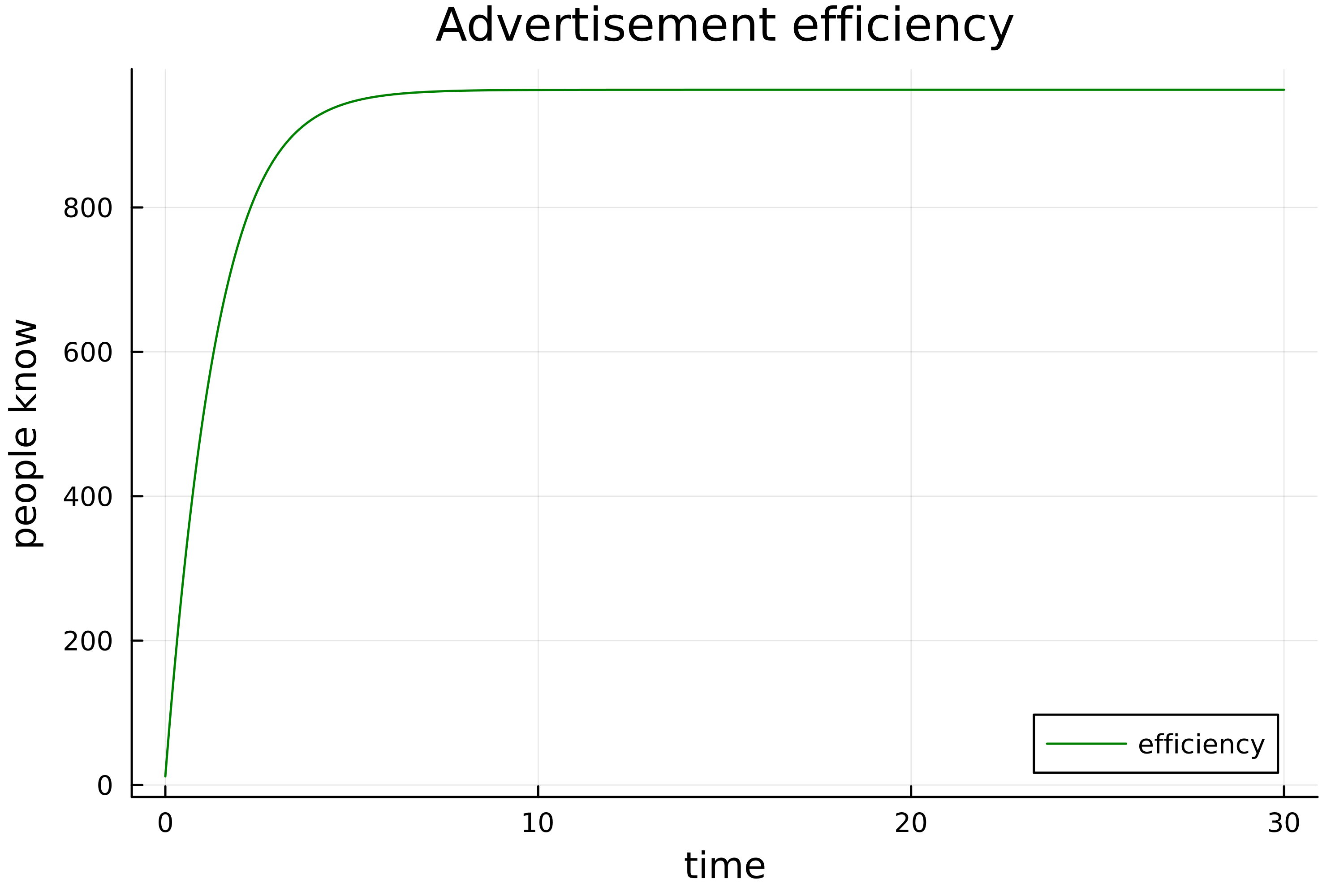


График эффективности рекламы для задачи 1.

## 3. Выполнение лабораторной для задачи 2.

### Код программы:

N = 963  
n0 = 12  
  
alpha = 0.00001  
beta = 0.35  
  
function ode(du, u, p, t)  
 du[1] = (alpha + beta \* u[1]) \* (N - u[1])  
end  
  
t\_arr = (0, 0.1)  
problem = ODEProblem(ode, [n0], t\_arr)  
solution = solve(problem, dtmax = 0.01)  
  
n = [u[1] for u in solution.u]  
T = [t for t in solution.t]  
  
  
function findmax\_dn(T, sol)  
 max\_dn = 0  
 max\_t = 0  
 max\_n = 0  
  
 for (i, t) in enumerate(T)  
 if sol(t, Val{1})[1] > max\_dn  
 max\_dn = sol(t, Val{1})[1]  
 max\_t = t  
 max\_n = n[i]  
 end  
 end  
 return max\_dn, max\_t, max\_n  
end   
  
max\_dn, max\_t, max\_n = findmax\_dn(T, solution)  
  
plt = plot(  
 dpi = 500,   
 title = "Advertisement efficiency",  
 xlabel = "time",  
 ylabel = "people know",  
 legend = true  
)  
  
plot!(  
 plt,  
 T,  
 n,  
 label = "Spreading",  
 color = :green  
)  
  
plot!(  
 plt,  
 [max\_t],  
 [max\_n],  
 seriestype = :scatter,  
 label = "Maximum popularity growth",  
 color = :black  
)  
  
savefig(plt, "./lab7/image/lab7\_2.png")

### График эффективности рекламы для задачи 2.

В итоге, получим вот такой график(рис. 2):

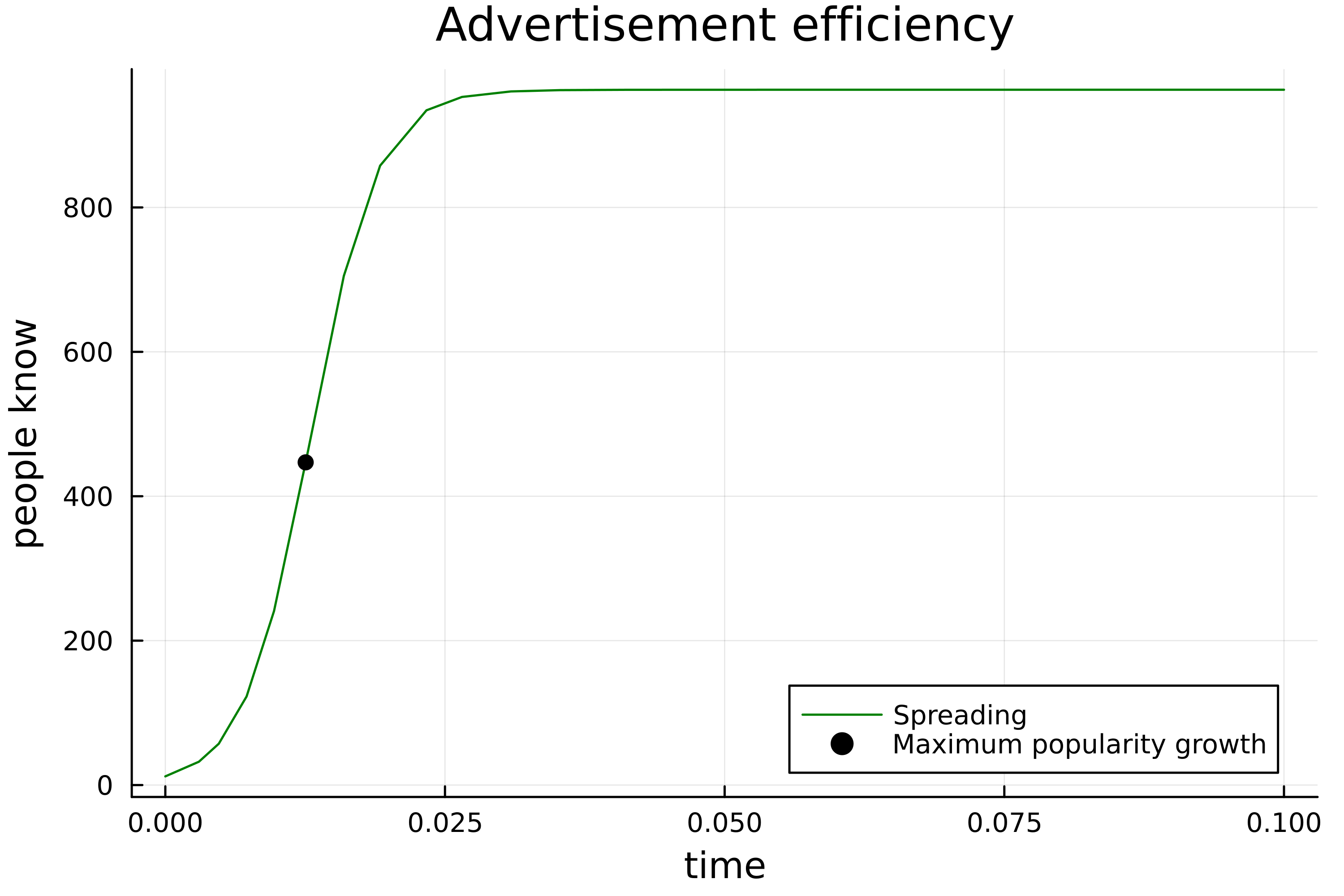


График эффективности рекламы для задачи 2.

## 4. Выполнение лабораторной для задачи 3.

### Код программы:

N = 963  
n0 = 12  
  
  
alpha = 0.51  
beta = 0.31  
  
function ode(du, u, p, t)  
 du[1] = (alpha\*sin(5\*t) + beta\*cos(3\*t)\*u[1]) \* (N - u[1])  
   
end  
  
t\_arr = (0, 0.1)  
problem = ODEProblem(ode, [n0], t\_arr)  
  
solution = solve(problem, dtmax = 0.01)  
  
n = [u[1] for u in solution.u]  
T = [t for t in solution.t]  
  
plt = plot(  
 dpi = 500,   
 title = "Advertisement efficiency",  
 xlabel = "time",  
 ylabel = "people know",  
 legend = true  
)  
  
plot!(  
 plt,  
 T,  
 n,  
 label = "efficiency",  
 color = :green  
)  
  
savefig(plt, "./lab7/image/lab7\_3.png")

### График эффективности рекламы для задачи 3.

В итоге, получим вот такой график(рис. 3):

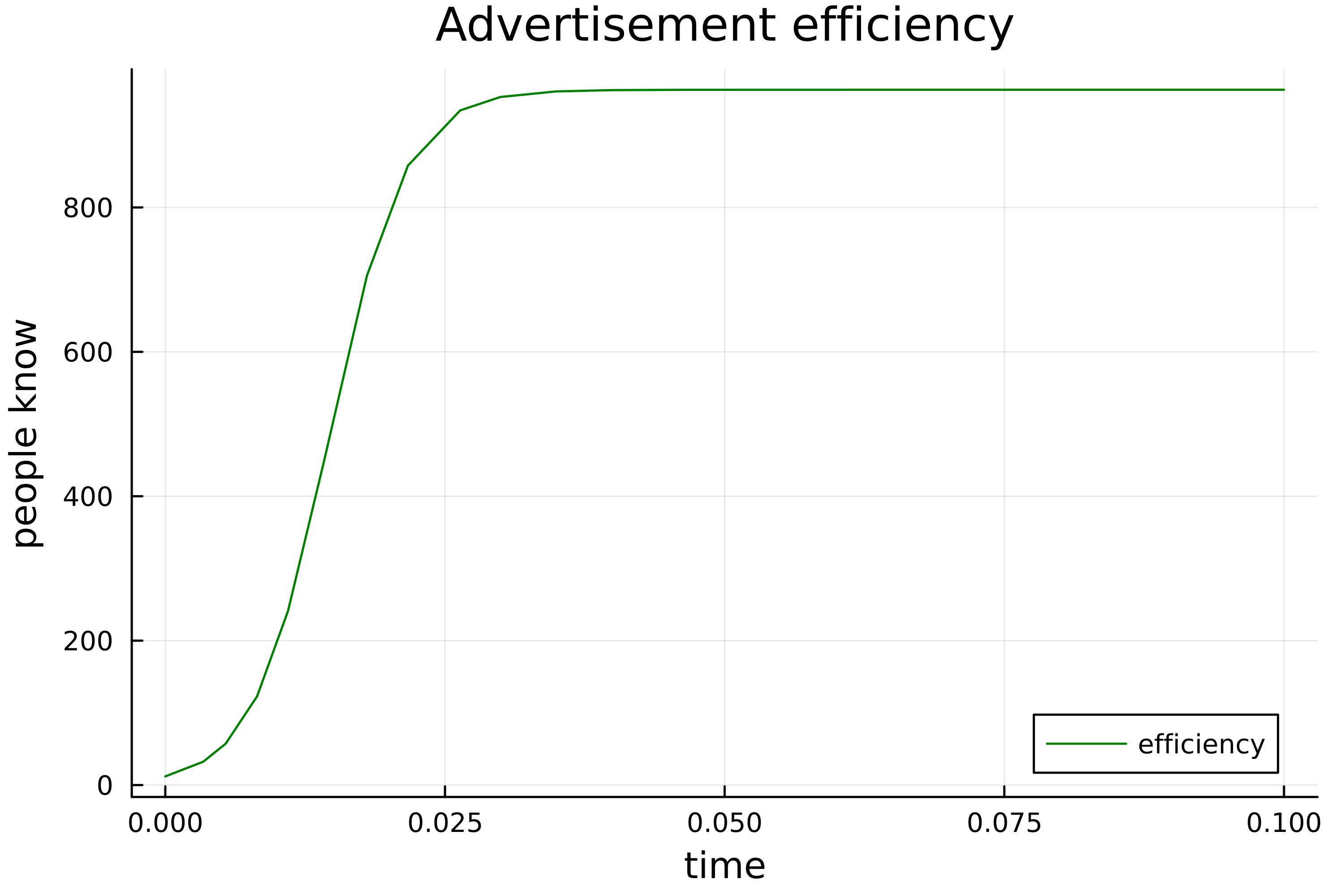


График эффективности рекламы для задачи 3.

# Выводы

В этой лабораторной работе мы изучили задачу об эффективности рекламы и подкрепили свои знания языка Julia и его библиотек.