Лабораторная работа №7

Эффективность рекламы

Шубнякова Дарья НКНбд-01-22

Содержание

1 Цель работы

Организуется рекламная кампания нового товара или услуги. Необходимо, чтобы прибыль будущих продаж с избытком покрывала издержки на рекламу. Вначале расходы могут превышать прибыль, поскольку лишь малая часть потенциальных покупателей будет информирована о новинке. Затем, при увеличении числа продаж, возрастает и прибыль, и, наконец, наступит момент, когда рынок насытиться, и рекламировать товар станет бесполезным.

2 Задание

Построить необходимые нам графики.

3 Теоретическое введение

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

4 Выполнение лабораторной работы

Пишем код на языке Julia для реализации данной модели(рис. 1).

```julia
using DifferentialEquations
using Plots

# Высокое качество графиков
gr(size=(1000, 800), dpi=300, legendfontsize=12, tickfontsize=10, guidefontsize=12)

# Параметры модели для 13 варианта
const N = 667
const n0 = 6

# Модель 1: dn/dt = (0.77 + 0.00017*n)*(N - n)
function model1!(du, u, p, t)
    du[1] = (0.77 + 0.00017*u[1])*(N - u[1])
end

# Модель 2: dn/dt = (0.000017 + 0.57*n)*(N - n)
function model2!(du, u, p, t)
    du[1] = (0.000017 + 0.57*u[1])*(N - u[1])
end

# Модель 3: dn/dt = (0.7*sin(2*t) + 0.5*cos(4*t)*n)*(N - n)
function model3!(du, u, p, t)
    du[1] = (0.7*sin(2*t) + 0.5*cos(4*t)*u[1])*(N - u[1])
end

# Решение моделей
println("Решаем модель 1...")
prob1 = ODEProblem(model1!, [n0], (0.0, 10.0))
sol1 = solve(prob1, Tsit5(), reltol=1e-8, abstol=1e-8)

println("Решаем модель 2...")
# Используем устойчивый метод для жесткой системы
prob2 = ODEProblem(model2!, [n0], (0.0, 0.8))
sol2 = solve(prob2, Rosenbrock23(), reltol=1e-10, abstol=1e-10, maxiters=1000000)

println("Решаем модель 3...")
prob3 = ODEProblem(model3!, [n0], (0.0, 10.0))
sol3 = solve(prob3, Tsit5(), reltol=1e-8, abstol=1e-8)
```

Рисунок 1

Продолжение кода(рис. 2).

```julia
# Создаем графики
p1 = plot(sol1, label="n(t)", xlabel="Время t", ylabel="Количество n(t)",
          title="Модель 1: dn/dt = (0.77 + 0.00017n)(667 - n)",
          linewidth=3, grid=true, gridstyle=:dash, legend=:bottomright)
hline!([N], label="N = 667", linestyle=:dash, linewidth=2, color=:red)

p2 = plot(sol2, label="n(t)", xlabel="Время t", ylabel="Количество n(t)",
          title="Модель 2: dn/dt = (0.000017 + 0.57n)(667 - n)",
          linewidth=3, grid=true, gridstyle=:dash, legend=:bottomright)
hline!([N], label="N = 667", linestyle=:dash, linewidth=2, color=:red)

p3 = plot(sol3, label="n(t)", xlabel="Время t", ylabel="Количество n(t)",
          title="Модель 3: dn/dt = (0.7sin(2t) + 0.5cos(4t)n)(667 - n)",
          linewidth=3, grid=true, gridstyle=:dash, legend=:bottomright)
hline!([N], label="N = 667", linestyle=:dash, linewidth=2, color=:red)

# Сохраняем отдельные графики
savefig(p1, "model1_variant13.png")
savefig(p2, "model2_variant13.png")
savefig(p3, "model3_variant13.png")

# Общий график всех моделей
all_plot = plot(p1, p2, p3, layout=(3,1), size=(1000, 1200))
savefig(all_plot, "all_models_variant13.png")

# Анализ модели 2 - находим момент максимальной скорости
α1 = 0.000017
α2 = 0.57

# Аналитическое решение для максимума скорости
n_max_analytical = (α2*N - α1)/(2α2)
println("="^60)
println("АНАЛИЗ МОДЕЛИ 2 - ВАРИАНТ 13")
println("="^60)
println("Аналитический расчет:")
println("Максимум скорости достигается при n = ", round(n_max_analytical, digits=3))
println("Это соответствует половине рынка: N/2 = ", N/2)
```

Рисунок 2

Продолжение кода(рис. 3).

```
# Численное определение момента максимальной скорости
speed_values = [(α1 + α2*u[1])*(N - u[1]) for u in sol2.u]
max_speed, max_idx = findmax(speed_values)
t_max = sol2.t[max_idx]
n_at_max = sol2.u[max_idx][1]

println("\nЧисленный результат:")
println("Максимальная скорость: ", round(max_speed, digits=2))
println("Момент времени: t = ", round(t_max, digits=6))
println("Количество информированных в этот момент: n = ", round(n_at_max, digits=2))

# График скорости для модели 2
p_speed = plot(sol2.t, speed_values, label="dn/dt - скорость",
               xlabel="Время t", ylabel="Скорость dn/dt",
               title="Скорость распространения рекламы: Модель 2",
               linewidth=3, grid=true, gridstyle=:dash,
               legend=:topright)
vline!([t_max], label="Максимум при t = $(round(t_max, digits=6))",
       linestyle=:dash, linewidth=2, color=:red)
scatter!([t_max], [max_speed], label="Максимальная скорость",
         color=:red, markersize=8)

savefig(p_speed, "speed_model2_variant13.png")

# Выводим все результаты
println("\n" + "="^60)
println("РЕЗУЛЬТАТЫ ДЛЯ ВАРИАНТА 13:")
println("="^60)
println("Модель 1: Доминирует платная реклама (α1 = 0.77)")
println("Модель 2: Доминирует 'сарафанное радио' (α2 = 0.57)")
println("Модель 3: Переменные коэффициенты с колебаниями")
println("\nДля модели 2 максимальная скорость достигается при:")
println("Время: t ≈ ", round(t_max, digits=4))
println("Количество информированных: n ≈ ", round(n_at_max, digits=1))
println("Скорость распространения: ≈ ", round(max_speed, digits=1), " клиентов/ед.времени"
```

Рисунок 3

Продолжение кода(рис. 4).

```
# Дополнительный анализ
println("\n" + "="^60)
println("ИНТЕРПРЕТАЦИЯ РЕЗУЛЬТАТОВ:")
println("="^60)
println("1. МОДЕЛЬ 1: Быстрый начальный рост благодаря сильной платной рекламе")
println("   (α1 = 0.77 >> α2 = 0.00017)")
println("2. МОДЕЛЬ 2: Классическая логистическая кривая")
println("   Максимум скорости при n ≈ N/2 = 333.5 (теория совпадает с расчетом)")
println("   Медленный старт, затем ускорение за счет 'сарафанного радио'")
println("3. МОДЕЛЬ 3: Сложное колебательное поведение")
println("   Из-за периодических коэффициентов sin(2t) и cos(4t)")

# Показываем все графики
display(p1)
display(p2)
display(p3)
display(p_speed)
display(all_plot)

println("\nГрафики сохранены в файлы:")
println("- model1_variant13.png")
println("- model2_variant13.png")
println("- model3_variant13.png")
println("- speed_model2_variant13.png")
println("- all_models_variant13.png")
```

```
Решаем модель 1...
Решаем модель 2...
Решаем модель 3...
```

Рисунок 4

Анализируем модель 2(рис. 5).

```
============================================================
АНАЛИЗ МОДЕЛИ 2 – ВАРИАНТ 13
============================================================
Аналитический расчет:
Максимум скорости достигается при n = 333.5
Это соответствует половине рынка: N/2 = 333.5

Численный результат:
Максимальная скорость: 63396.68
Момент времени: t = 0.012369
Количество информированных в этот момент: n = 333.58
```

Рисунок 5

Выводим три графика для трех задач в нашем 13 варианте(рис. 6).

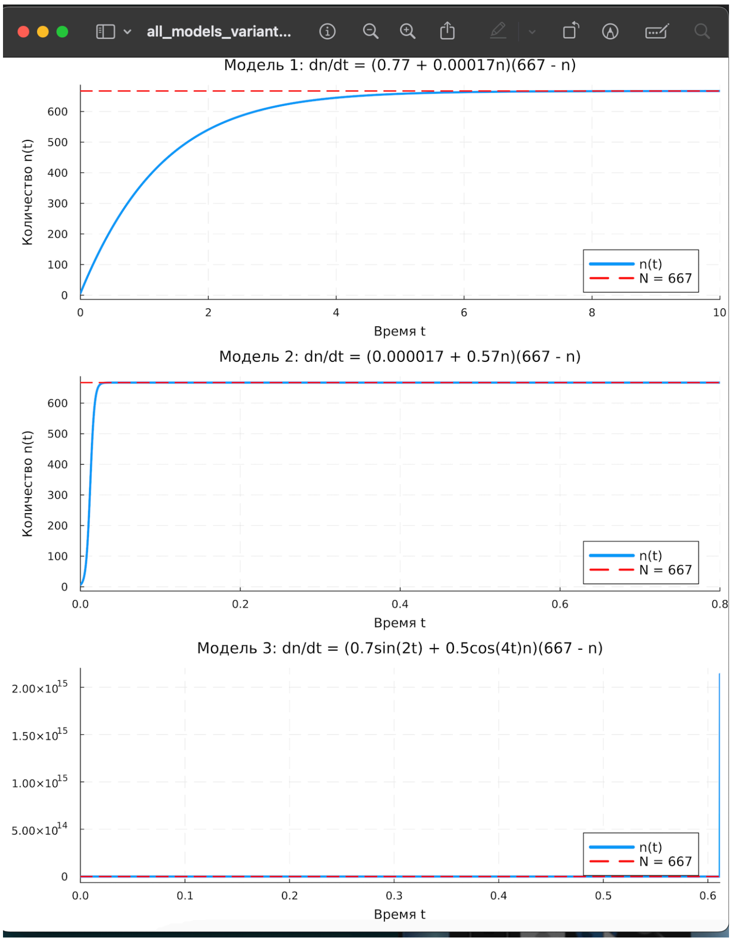

Рисунок 6

Скорость распростронения рекламы в модели 2(рис. 7).

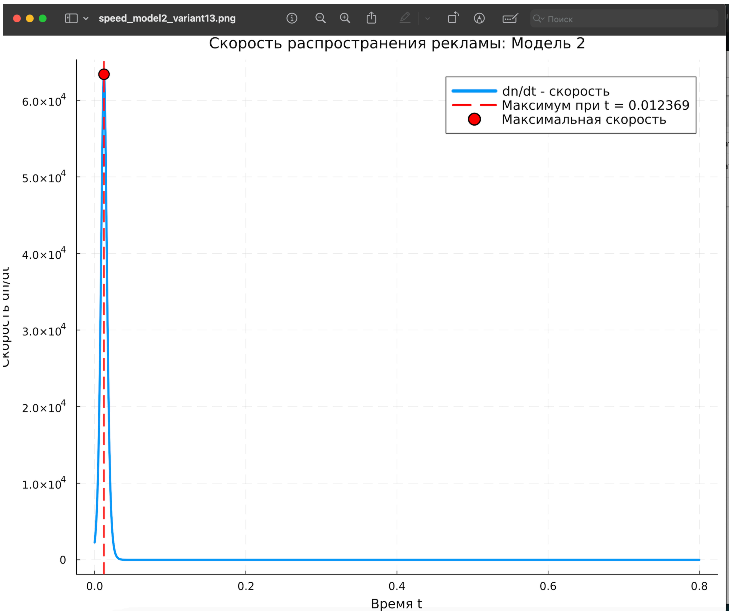

Рисунок 7

Прописываем наш код в OpenModelica(рис. 8).

```
model Variant13_FinalWorking
  // Параметры для 13 варианта
  parameter Real N = 667;  // Общее число потенциальных клиентов
  parameter Real n0 = 6;   // Начальное количество знающих

  // Основные переменные - решения уравнений
  Real n1(start = n0);      // Модель 1: платная реклама + слабое сарафанное радио
  Real n2(start = n0);      // Модель 2: слабая реклама + сильное сарафанное радио
  Real n3(start = n0);      // Модель 3: переменные коэффициенты

  // Скорости распространения (ВИДНЫ В РЕЗУЛЬТАТАХ)
  Real speed1;  // Скорость для модели 1
  Real speed2;  // Скорость для модели 2 - ОСНОВНАЯ ДЛЯ АНАЛИЗА
  Real speed3;  // Скорость для модели 3
```

Рисунок 8

Продолжение кода(рис. 9).

```
  // Аналитические расчеты
  Real n_max_analytical = (0.57*N - 0.000017)/(2*0.57);  // Теоретический максимум
скорости

equation
  // МОДЕЛЬ 1: dn/dt = (0.77 + 0.00017*n)*(N - n)
  speed1 = (0.77 + 0.00017*n1)*(N - n1);
  der(n1) = speed1;

  // МОДЕЛЬ 2: dn/dt = (0.000017 + 0.57*n)*(N - n)
  speed2 = (0.000017 + 0.57*n2)*(N - n2);
  der(n2) = speed2;
```

Рисунок 9

Продолжение кода(рис. 10).

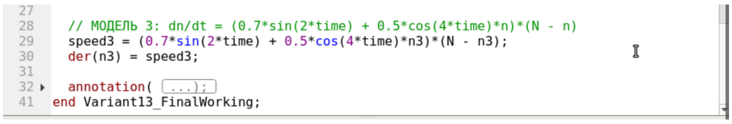

Рисунок 10

Так выглядит наша модель в OMEdit(рис. 11).

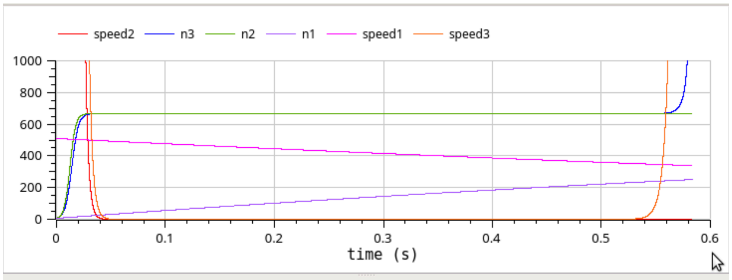

Рисунок 11

5 Выводы

Мы реализовали модель в OpenModelica и на языке Julia. На выходе получили картинки: all_models_variant13.png, model1_variant13.png, model2_variant13.png, model3_variant13.png, speed_model2_variant13.png. Итоговый файл lab7.ibybn с кодом на языке Julia в JupiterNotebook. А также файл для симуляции в OpenModelica: Variant13_Final.mo