

Лабораторная работа № 7

Тарусов Артём Сергеевич

2023, Москва

Целью данной работы является построение модели распространения рекламы.

Построить графики распространения рекламы для трех случаев. При этом объем аудитории $N = 810$, в начальный момент о товаре знает 11 человек. Для случая 2 определить, в какой момент времени скорость распространения рекламы будет иметь максимальное значение.

Опишем начальные значения согласно варианту 8 на языке Julia.

```
N ::Int64 = 810 # объем аудитории  
n0 ::Int64 = 11 # уже знающие о товаре люди
```

Рис. 1: Начальные значения на языке Julia

Опишем дифференциальное уравнение для первого случая.

```
function ode_fn(du, u, p, t)
    (n) = u
    du[1] = (0.64 + 0.00014*u[1])*(N - u[1])
end
```

Рис. 2: Дифференциальное уравнение для первого случая на языке Julia

Получим решение дифференциального уравнения.

```
v0 :: Array{Int64} = [n0]
tspan = (0.0, 30.0)
prob = ODEProblem(ode_fn, v0, tspan)
sol = solve(prob, dtmax = 0.05)
n = [u[1] for u in sol.u]
T = [t for t in sol.t]
```

Рис. 3: Решение дифференциального уравнения для первого случая на языке Julia

Построим график распространения рекламы для первого случая.

```
plt = plot(  
    dpi = 300,  
    title = "Эффективность рекламы ",  
    legend = false)  
plot!(  
    plt,  
    T,  
    n,  
    color = :blue)  
  
savefig(plt, "out/lab07_1.png")
```

Рис. 4: Построение графика распространения рекламы для первого случая на языке Julia

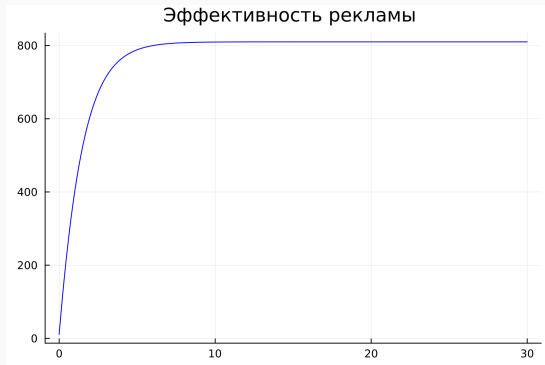


Рис. 5: График распространения рекламы для первого случая, построенный на Julia

Изменим дифференциальное уравнение для второго случая.

```
function ode_fn(du, u, p, t)
    (n) = u
    du[1] = (0.000014 + 0.63*u[1])*(N - u[1])
end
```

Рис. 6: Дифференциальное уравнение для второго случая на языке Julia

Определим, в какой момент времени скорость распространения рекламы будет иметь максимальное значение.

```
max_dn ::Int64 = 0;
max_dn_t ::Int64 = 0;
max_dn_n ::Int64 = 0;
for (i, t) in enumerate(T)
    if sol(t, Val{1})[1] > max_dn
        global max_dn = sol(t, Val{1})[1]
        global max_dn_t = t
        global max_dn_n = n[i]
    end
end
```

Рис. 7: Определение момента времени, когда скорость распространения рекламы будет иметь максимальное значение, на языке Julia

По аналогии с предыдущим построением получим график для второго случая.

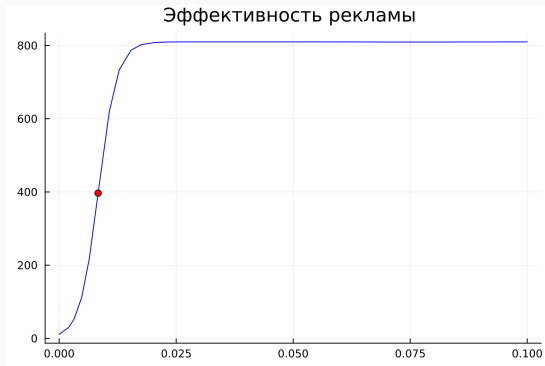


Рис. 8: График распространения рекламы для второго случая, построенный на Julia

Изменим дифференциальное уравнение для третьего случая.

```
function ode_fn(du, u, p, t)
    (n) = u
    du[1] = (0.7 + 0.4*cos(t)*u[1])*(N - u[1])
end
```

Рис. 9: Дифференциальное уравнение для третьего случая на языке Julia

Получим получим график для третьего случая.

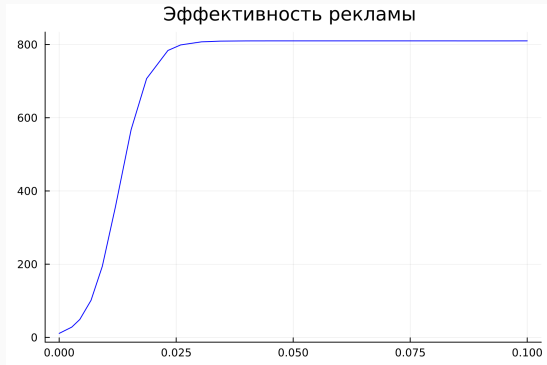


Рис. 10: График распространения рекламы для третьего случая, построенный на Julia

Построим модель для первого случая на языке OpenModelica.

```
1  model lab07_1
2  Real N = 810;
3  Real n;
4  initial equation
5  n = 11;
6  equation
7  der(n) = (0.64 + 0.00014*n) * (N-n) ;
8  end lab07_1;
```

Рис. 11: Построение модели для первого случая на языке OpenModelica

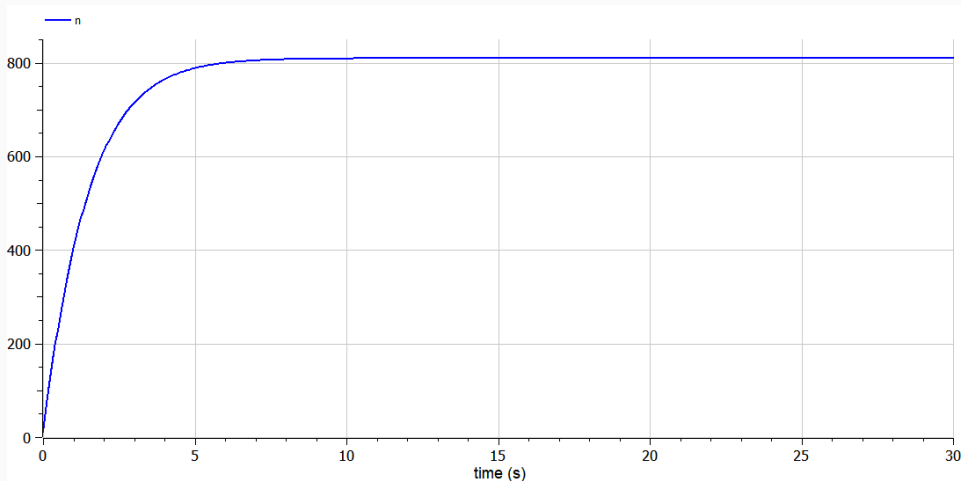


Рис. 12: График распространения рекламы для первого случая, построенный на языке OpenModelica

Построим модель для второго случая на языке OpenModelica. Находить, в какой момент времени скорость распространения рекламы будет иметь максимальное значение, не будем, так как реализовать это базовыми средствами OpenModelica довольно затруднительно.

```
1 model lab07_2
2   Real N = 810;
3   Real n;
4   initial equation
5     n = 11;
6   equation
7     der(n) = (0.000014 + 0.63*n) * (N-n) ;
8 end lab07_2;
```

Рис. 13: Построение модели для второго случая на языке OpenModelica

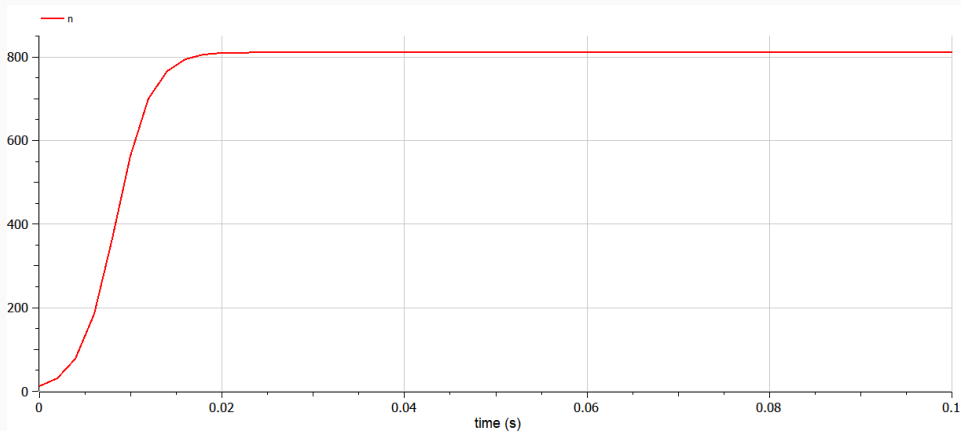


Рис. 14: График распространения рекламы для второго случая, построенный на языке OpenModelica

Построим модель для третьего случая на языке OpenModelica.

```
1 model lab07_3
2 Real N = 810;
3 Real n;
4 initial equation
5 n = 11;
6 equation
7 der(n) = (0.7 + 0.4*cos(time)*n) * (N-n);
8 end lab07_3;
```

Рис. 15: Построение модели для третьего случая на языке OpenModelica

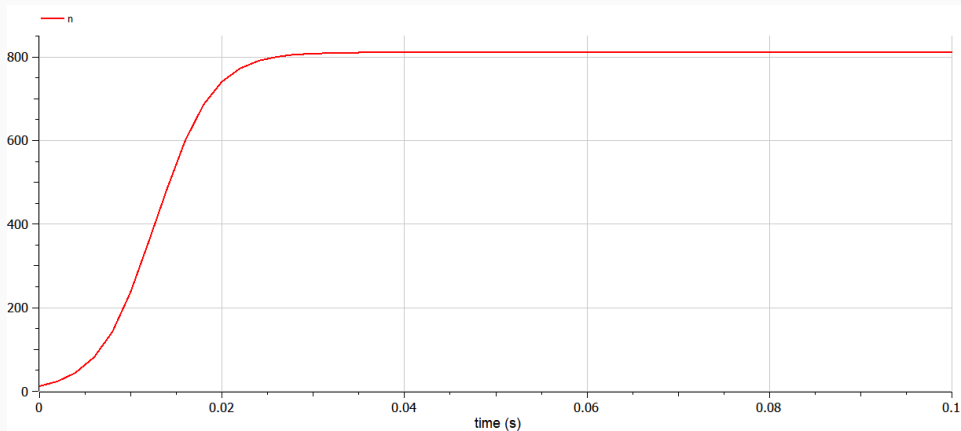


Рис. 16: График распространения рекламы для третьего случая, построенный на языке OpenModelica

В итоге проделанной работы мы построили графики распространения рекламы для трех случаев на языках Julia и OpenModelica. Построение модели распространения рекламы на языке OpenModelica занимает значительно меньше строк, чем аналогичное построение на Julia. Кроме того, построения на языке OpenModelica проводятся относительно значения времени t по умолчанию, что упрощает нашу работу. Но при этом вычисление момент времени, когда скорость распространения рекламы будет иметь максимальное значение, довольно затруднительно на OpenModelica.