Model gradacji owadów

Bartosz Zbik

2024-05-18

Wyjściowe równanie

$$\frac{dN}{dt} = r_0 N \left(1 - \frac{N}{K}\right) - \frac{BN^2}{A^2 + N^2} \tag{1}$$

możemy zredukować (przy pomocy linowych transformacji) do

$$\frac{dn}{d\tau} = rn \left(1 - \frac{n}{k}\right) - \frac{n^2}{1 + n^2}, \tag{2}$$

gdzie

$$n = \frac{N}{A}, \quad \tau = \frac{B}{A}t, \quad r = \frac{A}{B}r_0, \quad k = \frac{1}{A}k. \tag{3}$$

1 Kod

```
using DifferentialEquations
using CairoMakie
set_theme!(theme_latexfonts())


# p = (k, r)
function model(n, p, t)
    return p[2] * n * (1.0 - n / p[1]) - n^2 / (1 + n^2)
end


function findsol(k, r, x0)
    tspan = (0.0, 50.0)
    prob = ODEProblem(model, x0, tspan, (k, r))
    sol = solve(prob)
    return sol
end

function make_plot(k, r, x0)
    sol = findsol(k, r, x0)
    tspan = 0..50

    fig = Figure()
    ax = Axis(fig[1, 1])
    lines!(ax, tspan, t -> sol(t))
    ax.title = L"k = %$k,~~ r = %$r,~~ x_0 = %$(x0)"
    ax.xlabel = "time"
    ax.ylabel = "population density"
    ax.limits = (nothing, (0.0, nothing))

   return fig
end

# k = 8
k8_x0_rng = [0.5, 1.0, 3.0, 5.0]
```

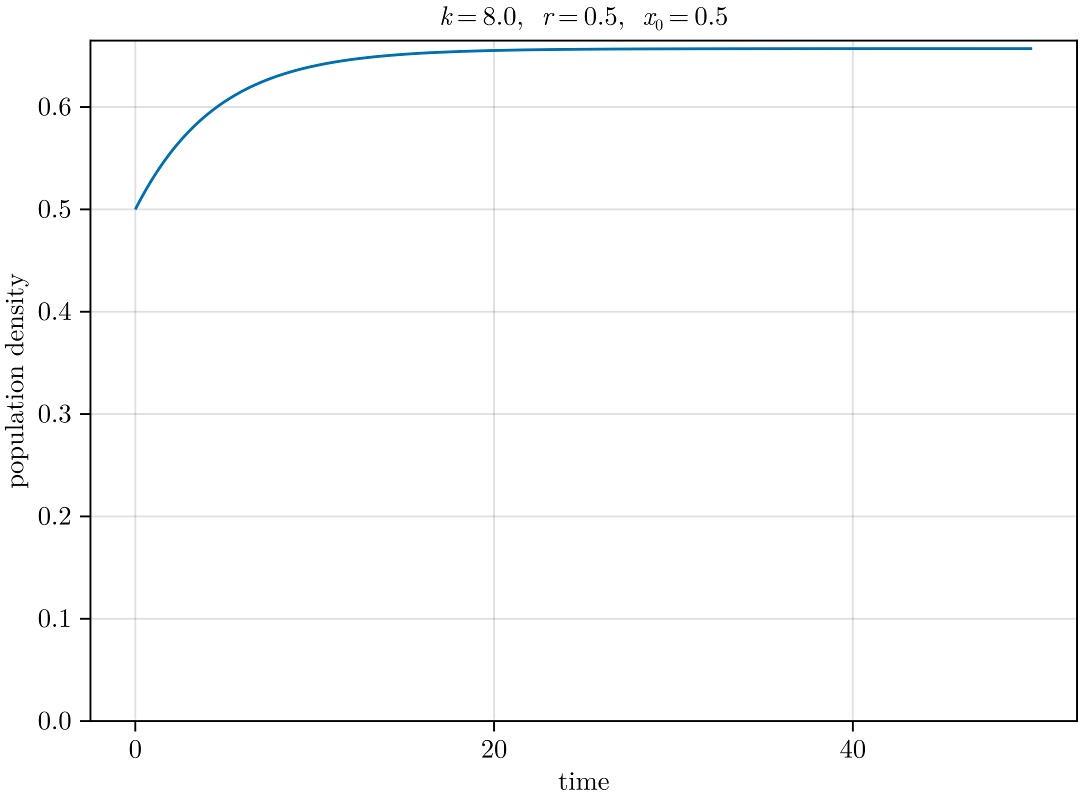
```julia
for i in eachindex(k8_x0_rng)
    save("out/model_k8_$i.pdf", make_plot(8.0, 0.5, k8_x0_rng[i]))
end

# k = 5
k5_x0_rng = [0.5, 1.0]
for i in eachindex(k5_x0_rng)
    save("out/model_k5_$i.pdf", make_plot(5.0, 0.5, k5_x0_rng[i]))
end
```

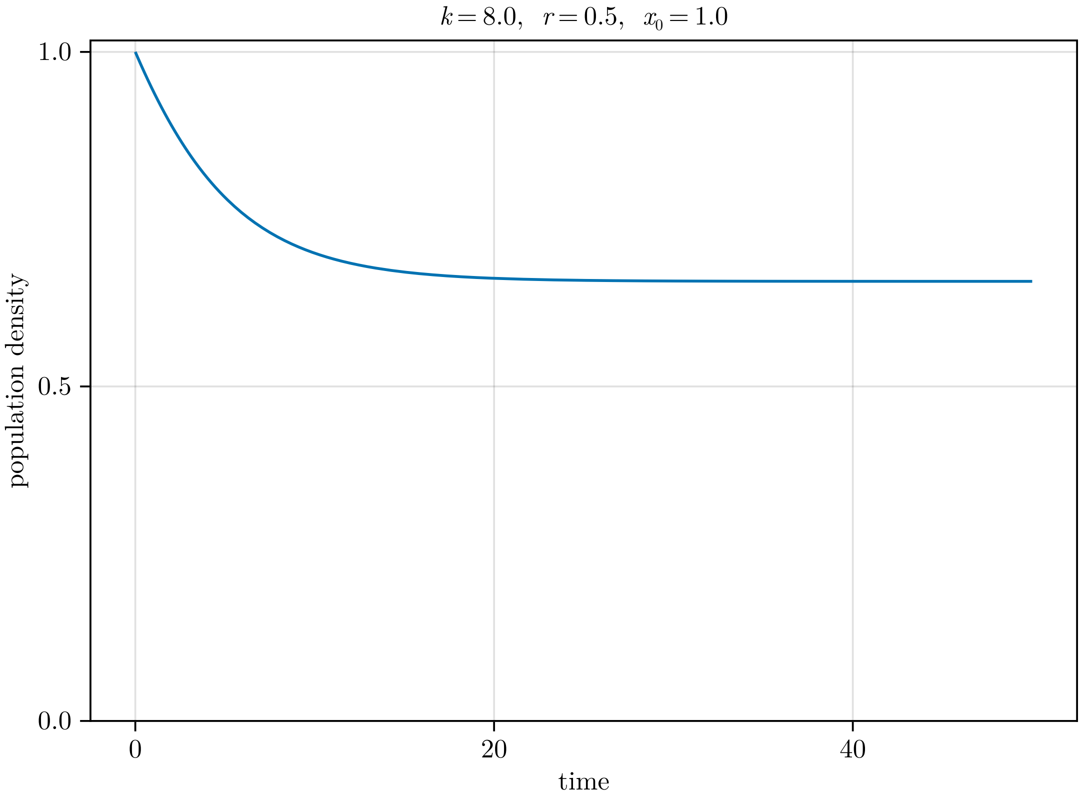
2 Wizualizacja wyników

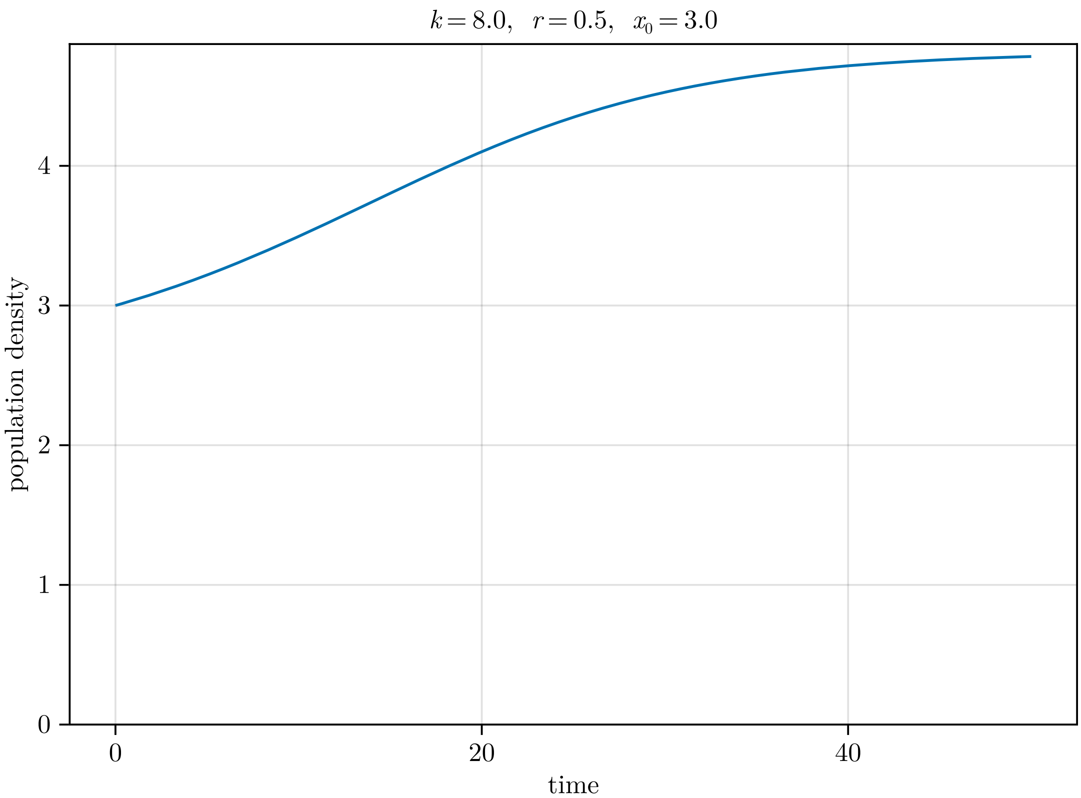
We wszystkich przypadkach układ dążył do stanu stacjonarnego.

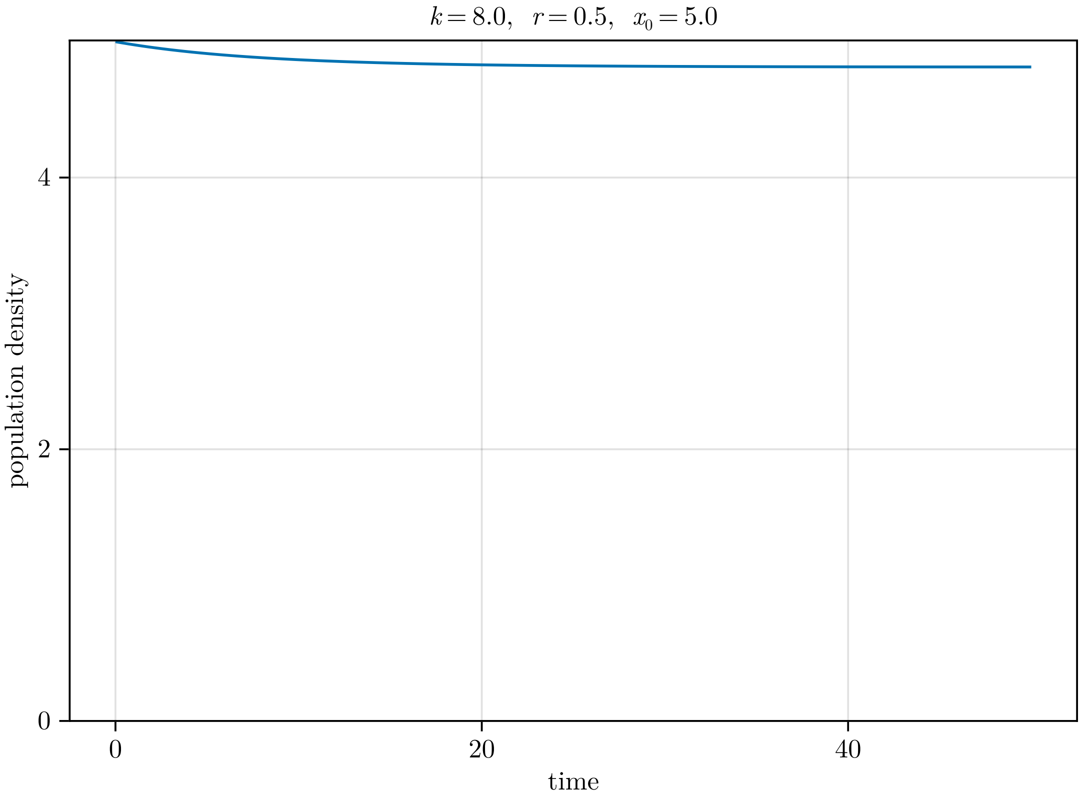
2.1 Rozwiązania dla $k = 8$

Dla $k = 8$ można zauważyć, że są co najmniej dwa stabilne stany stacjonarne. Jeden dla n^* gdzieś pomiędzy 0.6, a 0.7. Drugi dla n^* gdzieś pomiędzy 4.5, a 5.0.




$k = 8.0, \quad r = 0.5, \quad x_0 = 1.0$
1.0
0.5
0.0
population density
0
20
40
time


$k = 8.0, \quad r = 0.5, \quad x_0 = 3.0$
4
3
2
1
0
population density
0
20
40
time



$k = 8.0, \quad r = 0.5, \quad x_0 = 5.0$
population density
0
2
4
0
20
40
time

2.2 Rozwiązania dla $k = 5$