Diagram Bifurkacyjny

Bartosz Zbik

2024-03-17

Moje rozwiązanie korzysta z faktu, że badane równanie

$$\dot{x} = rx + x^3 - x^5 \tag{1}$$

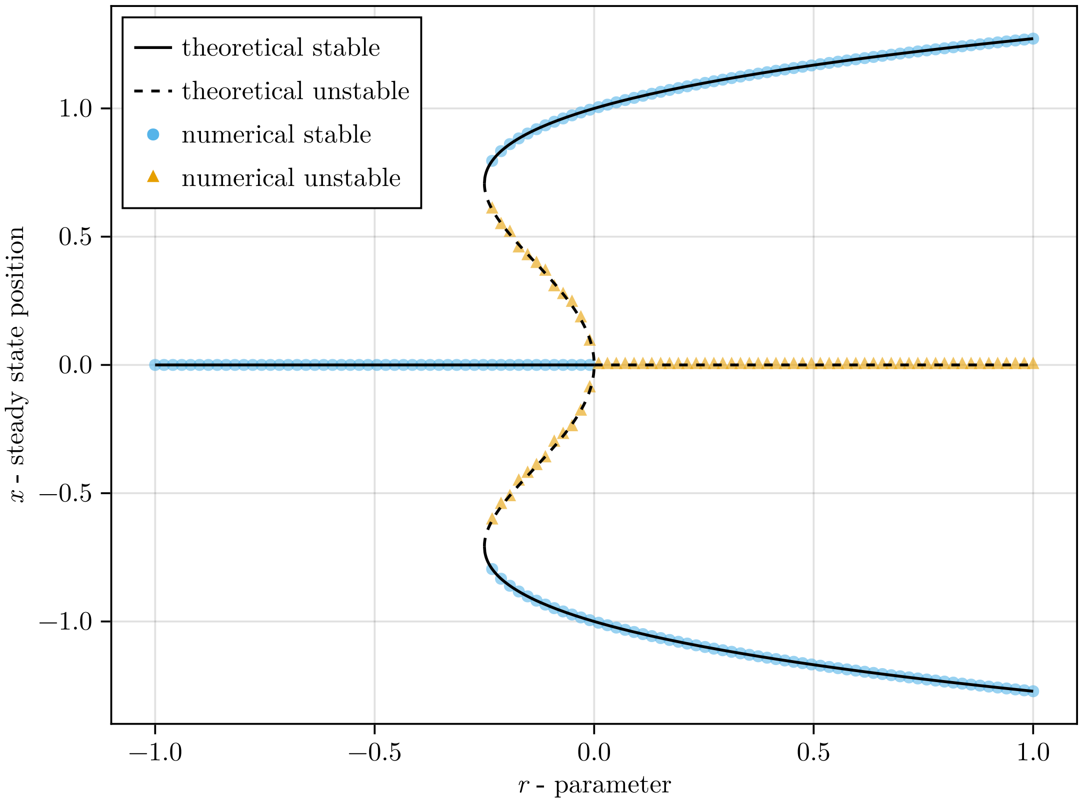
nie ma rozwiązań uciekających do $\pm\infty$.

1 Kod

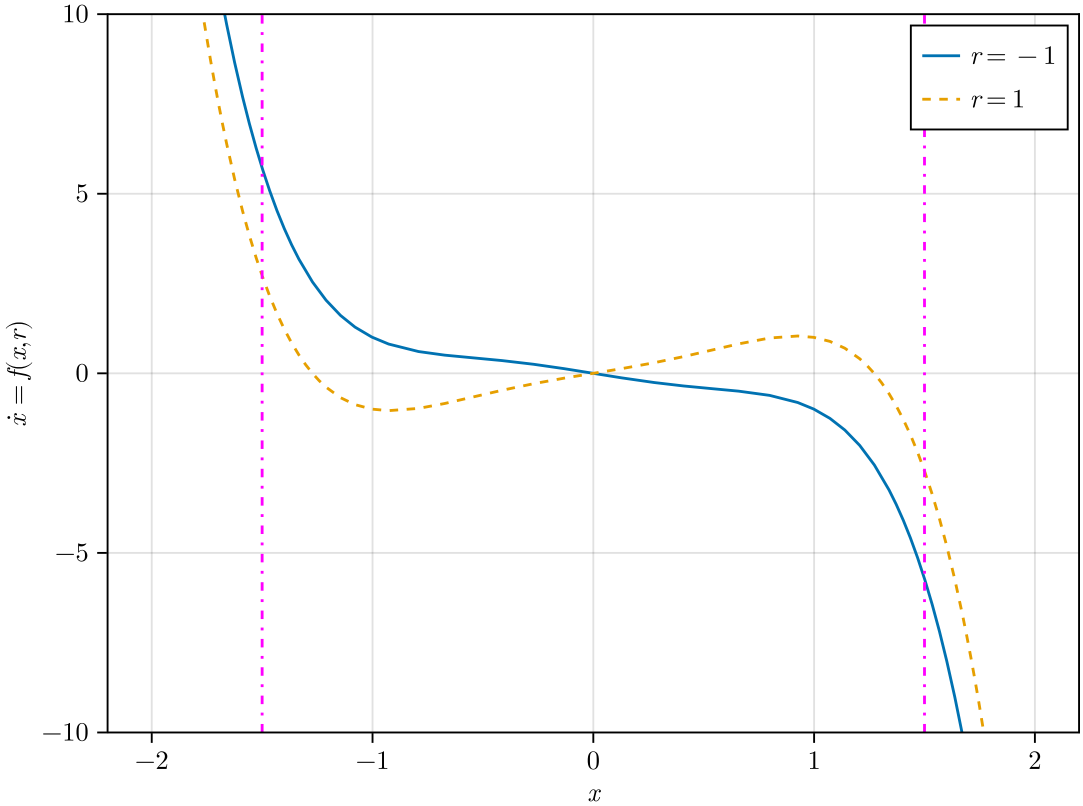
```
using DifferentialEquations, Statistics

# define the differential equation
f(x, r, t) = r * x + x^3 - x^5


# r < -1 is uninteresting as it gives only one equilibrium
# r > 1 results in 3 equilibrium points which properties are all the same
# Therefore only -1 < r < 1 is considered
r_rng = range(-1, 1, length=100)

# x(0) ranging from -1.5 to 1.5 will cover all interesting cases
# which follows from the plots of f(x, -1) and f(x, 1)
x0_rng = range(-1.5, 1.5, length=100)

# uniform norm of f(x, r) on the interesting set is less then 10
# thus choosing the time step around 10^-2 will be sufficient to approximate the
     solution
# which fastest decay is slower then exp(-10 * x) (characteristic time scale
    0.1)
sup_f = abs(f(1.5, -1, 0)) # 5.71875
dt = 1e-2

function final_state(f, r, x0, dt, tmax)
    # define the Problem object
    prob = ODEProblem(f, x0, (0, tmax), r)
    # solve using Runge-Kutta fourth order method
    # save_everystep=false drops intermediate points (we are only interested in
        the final position)
    sol = solve(prob, RK4(), dt=dt, adaptive=false, save_everystep=false,
        maxiters = 1e7)
    return sol.u[end]
end

function find_steady_states(r, f, x0_rng, dt, tmax, atol)
    # solves the equation to get final positions after a long time (tmax)
    xf_rng = [final_state(f, r, x0, dt, tmax) for x0 in x0_rng]

    # finds neighbouring solutions that converge to different attractors
    # is true if x0 -> a and x0 + dx -> b where a and b are different |a-b| >
        atol
    unst_index = findall([!isapprox(xf_rng[i], xf_rng[i+1], atol=atol) for i in
        1:length(x0_rng)-1])

    # unstable steady states are those found above
    unstable = [0.5 * (x0_rng[i] + x0_rng[i+1]) for i in unst_index]

    # unstable steady states divide the x plane to regions attracted to
        different stable steady states
```

```
    # this groups xf_rng by the attractor to which the solution approaches and
        calculates the mean xf
    border_index = [0, unst_index..., length(xf_rng)]
    stable = [mean(xf_rng[border_index[i]+1:border_index[i+1]]) for i in 1:
        length(border_index)-1]
    return stable, unstable
end


# those values are guessed but they work
tmax = 2e2
atol = 1e-2
# find steady states
all_ss = [find_steady_states(r, f, x0_rng, dt, tmax, atol) for r in r_rng]

# make plots for the report
include("./make_plots.jl")
```

2 Wizualizacja wyników



Rysunek 1: Diagram bifurkacyjny wyznaczony numerycznie (przez rozwiązywanie równania metodą Rungego-Kutty) porównany z analitycznym rozwiązaniem problemu bifurkacji w układzie.



Rysunek 2: Zależność $\dot{x}$ od x dla skrajnych badanych wartości r.