

Porównanie różnych metod numerycznego rozwiązywania liniowego układu równań różniczkowych

Bartosz Zbik

2024-03-02

1 Kod

```
# imports https://docs.sciml.ai/DiffEqDocs/stable/
using DifferentialEquations

using LinearAlgebra

#####
# Explicit methods
#####

# define the differential equation
# X - current system state
# p - parameter vector
# t - time
f(X, p, t) = [1023.0 2023.0; (-1024.0) (-2024.0)] * X

# initial conditions
X0 = [1.0, 0.0]
# time domain on which to solve the problem
tspan = (0.0, 0.125)

# specifies the "problem". It's specific for this ecosystem.
prob = ODEProblem(f, X0, tspan)

# Solve using different algorithms: https://docs.sciml.ai/DiffEqDocs/stable/
# solvers/ode_solve/
# Euler(), RK4() - specifies the algorithm
sol_a = solve(prob, Euler(), dt=1 / 2048)
sol_b = solve(prob, Euler(), dt=1 / 512)
sol_c = solve(prob, Euler(), dt=1 / 256)

sol_f = solve(prob, RK4(), dt=1 / 512, adaptive=false)
sol_g = solve(prob, RK4(), dt=1 / 256, adaptive=false)

#####
# Non Explicit methods
#####

# Non explicit Euler method for a linear equation
#  $x_{n+1} = x_n + dt * A x_n$ 
#  $(1 - A dt) x_{n+1} = x_n$ 
#  $x_{n+1} = (1 - A dt)^{-1} x_n$ 
function BackwardEulerStep(x, A, dt)
    # below algebraic operations work on matrices and vectors
    # x is a column vector
    # A is 2x2 matrix
    # [1 0; 0 1] is a 2x2 identity matrix
    return ([1 0; 0 1] - A * dt)^-1 * x
end

A = [1023.0 2023.0; (-1024.0) (-2024.0)]
```

```

nsteps_d = floor(Int, 1 + tspan[2] / (1 / 512))
sol_d = zeros(1 + nsteps_d, 2)
sol_d[1, :] .= X0
for i in 2:nsteps_d+1
    sol_d[i, :] .= BackwardEulerStep(sol_d[i-1, :], A, 1 / 512)
end
time_d = collect(0:1/512:0.125+1/512)

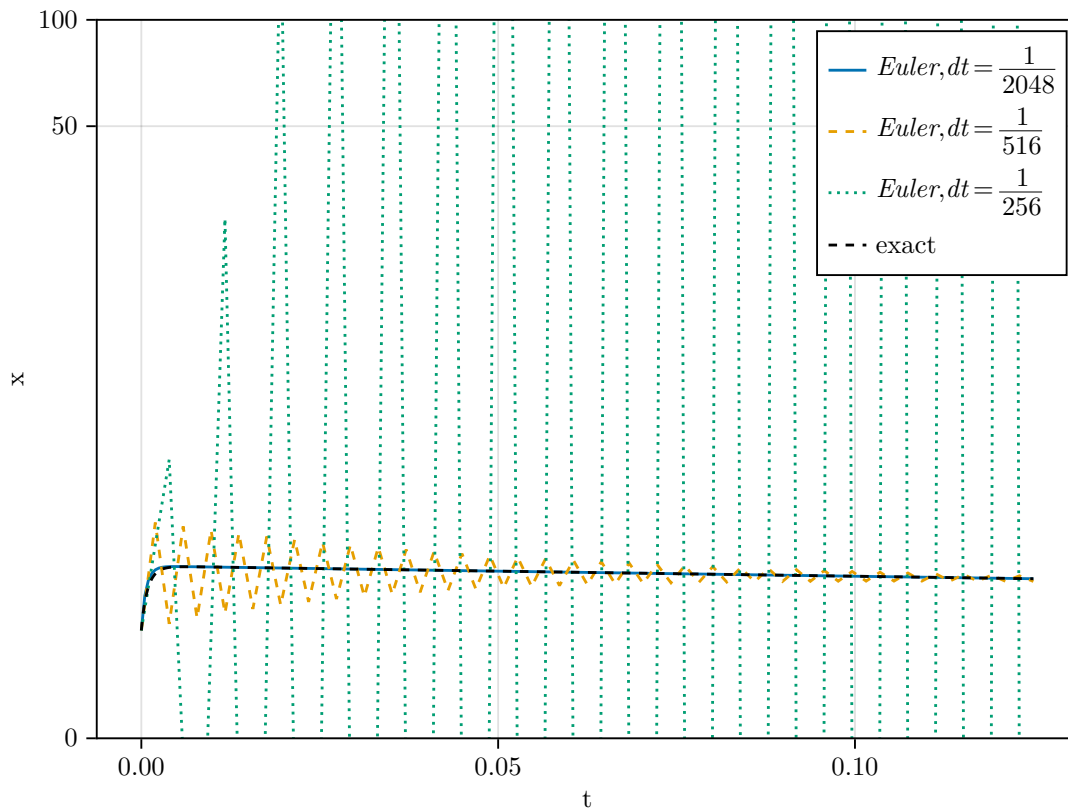
nsteps_e = floor(Int, 1 + tspan[2] / (1 / 256))
sol_e = zeros(1 + nsteps_e, 2)
sol_e[1, :] .= X0
for i in 2:nsteps_e+1
    sol_e[i, :] .= BackwardEulerStep(sol_e[i-1, :], A, 1 / 256)
end
time_e = collect(0:1/256:0.125+1/256)

function exact_sol(A, t, X0)
    return exp(A * t) * X0
end

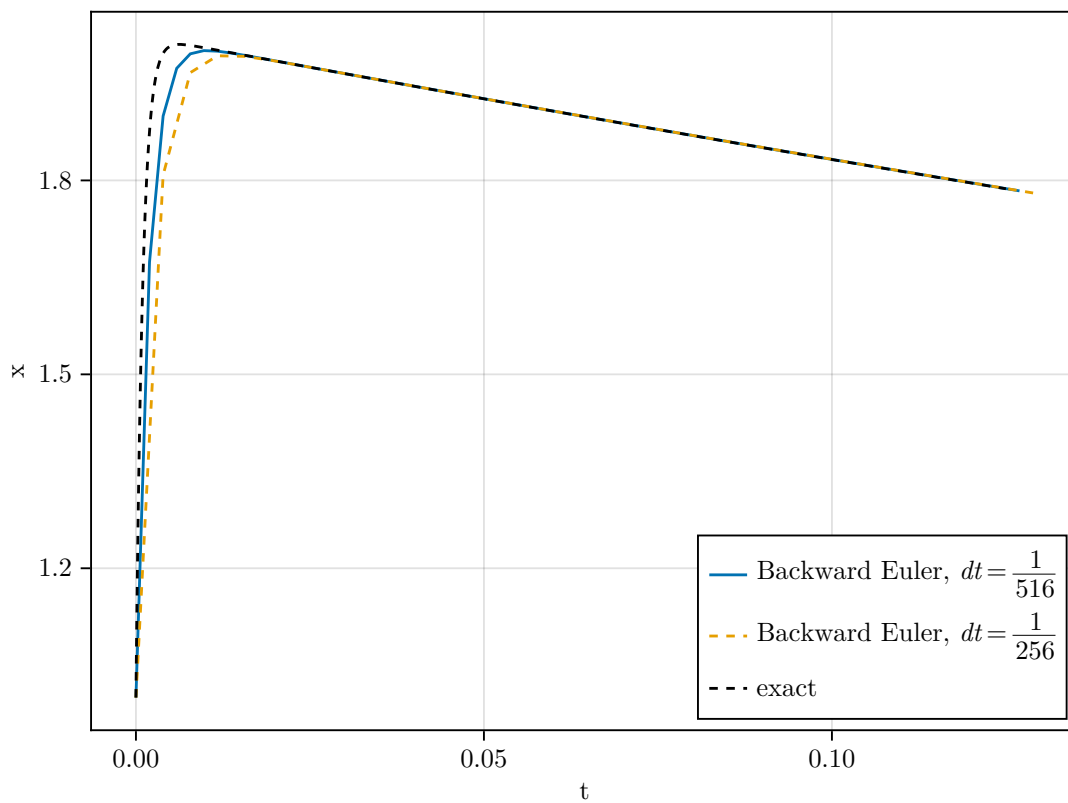
# Makes plots for the report
include("make_plots.jl")

```

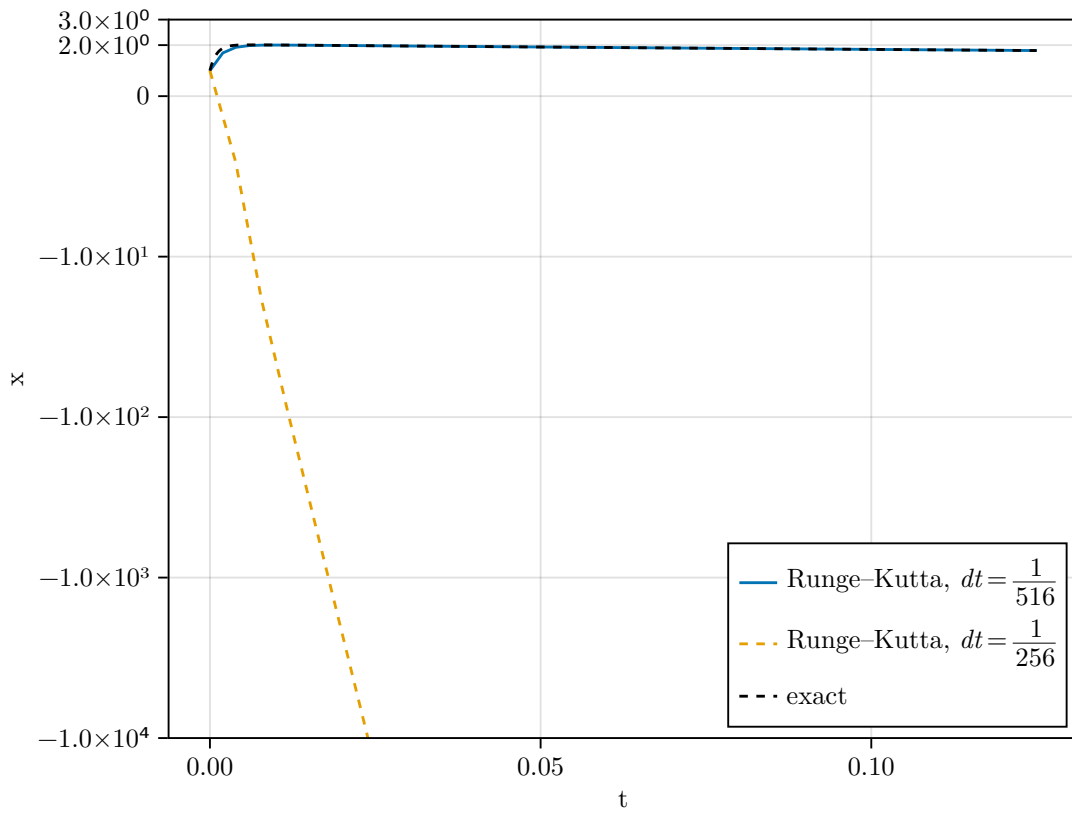
2 Wizualizacja wyników



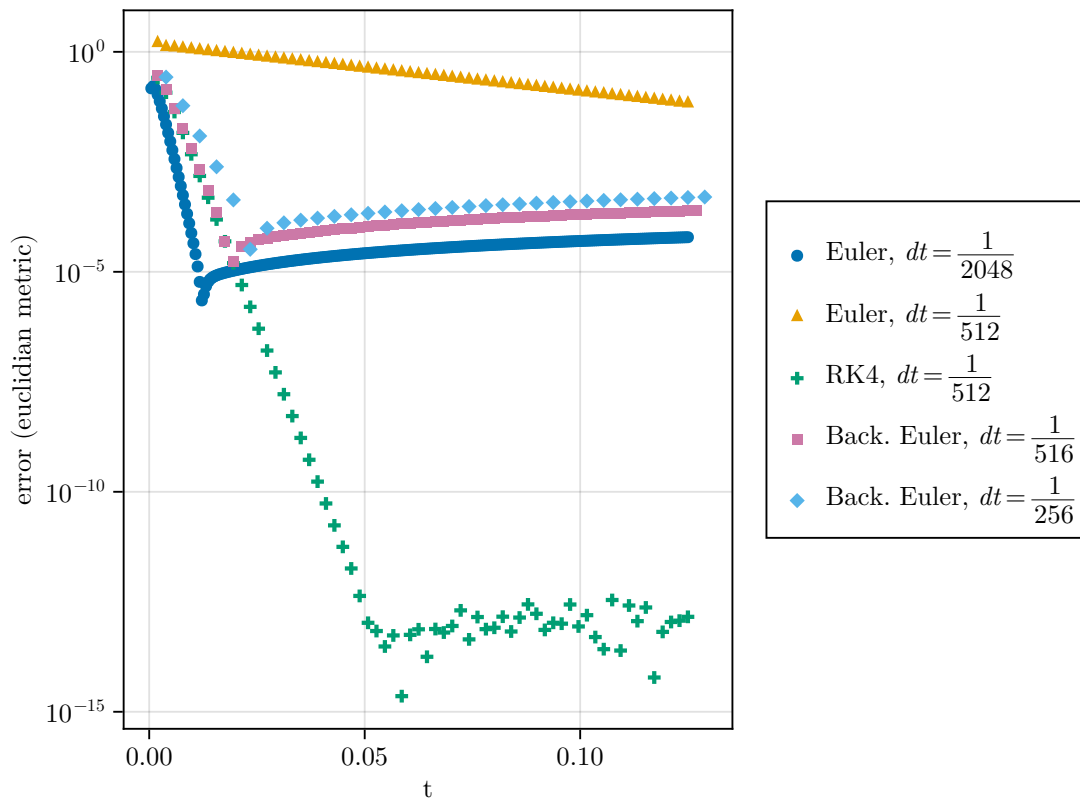
Rysunek 1: Jawna metoda Eulera z różnym krokiem czasowym.



Rysunek 2: Niejawna metoda Eulera z różnym krokiem czasowym. Tutaj konieczne jest zainwestowanie pewnej dodatkowej wiedzy o równaniu (trzeba odwrócić relację między x_{n+1} , a x_n).



Rysunek 3: Czterokrokowa metoda Rungego-Kutty ze stałym krokiem czasowym (nieadaptowalna)



Rysunek 4: Porównanie błędów numerycznych jakimi obarczone były rozwiązania poszczególnymi metodami.