Vaccination optimisation on an SIR model using JuMP.jl

Initial version here by Simon Frost (@sdwfrost)
Current version Sandra Montes (@slmontes), 2025-03-10

Introduction

This example examines the optimal control of an SIR model through vaccination, which reduces the number of susceptible individuals according to the following set of equations:

$$
\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{d}t} &= -\beta S I - \upsilon(t) S, \\
\frac{\mathrm{d}I}{\mathrm{d}t} &= \beta S I - \gamma I, \\
\frac{\mathrm{d}C}{\mathrm{d}t} &= \beta S I
\end{aligned}
$$

Similar to our previous examples, the population is divided into three categories: susceptible individuals (`S`), infected individuals (`I`), and the total number of cases (`C`). However in this case, Susceptible individuals are vaccinated at a per-capita rate `υ(t)`.

The optimal control problem is defined as the policy that minimises the total number of cases (i.e., the final size of the epidemic) while adhering to the following constraints: (a) the vaccination rate, `υ`, cannot exceed a maximum value, indicating a limit on the rate of vaccination, and (b) there is a cost associated with the vaccination process, measured as the integral of `υ(t)*S(t)` over time, which cannot exceed a predetermined level. Again, we determine the optimal policy numerically using a simple Euler discretisation and then JuMP.jl with IPOPT to optimise.

Libraries

```
using OrdinaryDiffEq
using DiffEqCallbacks
using JuMP
using Ipopt
using Plots
using DataInterpolations
using NonlinearSolve;
```

Functions

ODE system

```
function sir_ode!(du,u,p,t)
    (S, I, C) = u
    (β, γ, υ) = p
    @inbounds begin
        du[1] = -β*S*I - υ*S
        du[2] = β*S*I - γ*I
        du[3] = β*S*I
    end
    nothing
end;
```

Running the model without intervention

Parameters

```
u0 = [0.99, 0.01, 0.0]; #S, I, C (cumulative incidence)
p = [0.5, 0.25, 0]; # β, γ, υ
```

```
t0 = 0.0
tf = 100
dt = 0.1
ts = collect(t0:dt:tf)
alg = Tsit5();
```