‘Flattening the curve’ 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 explores the optimal control of an SIR (Susceptible-Infected-Recovered) model using a time-varying intervention that reduces the infection rate, similar to the lockdown example.

The following differential equations also describe this model :

$$\begin{aligned}
\frac{\mathrm{d}S}{\mathrm{dt}} &= -\beta(1-\upsilon(t))SI, \\
\frac{\mathrm{d}I}{\mathrm{dt}} &= \beta(1-\upsilon(t))SI - \gamma I, \\
\frac{\mathrm{d}C}{\mathrm{dt}} &= \beta(1-\upsilon(t))SI,
\end{aligned}$$

In this case, the optimal control problem is formulated to minimise the total intervention cost, measured as the integral of `υ(t)` over time while ensuring that the number of infected individuals, `I`, stays below a set threshold `I_max`. This constraint is introduced to achieve the objective of ‘flattening the curve,’ meaning that the optimal intervention policy `υ(t)` balances the cost of intervention with the need to keep the infection spread manageable, ensuring that the number of infected individuals never exceeds the threshold `I_max`. 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*(1-υ)*S*I
        du[2] = β*(1-υ)*S*I - γ*I
        du[3] = β*(1-υ)*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();
```