Vaccination optimisation on an SIR model using JuMP.jl

Initial version [here](https://github.com/epirecipes/sir-julia/blob/master/markdown/function_map_vaccine_jump/function_map_vaccine_jump.md) 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:

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();

Using ODEProblem

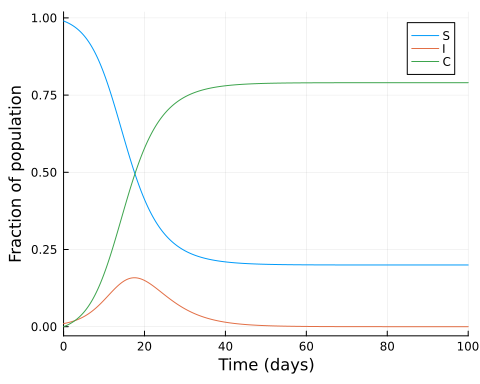
prob1 = ODEProblem(sir\_ode!, u0, (t0, tf), p)  
sol1 = solve(prob1, alg, saveat=ts);

Without control the peak fraction of infected individuals is

peak\_value, peak\_index = findmax(sol1[2, :])   
println("The maximum fraction of infected at a `dt` time is: ", peak\_value)

The maximum fraction of infected at a `dt` time is: 0.15845528864997238

plot(sol1,  
 xlim=(0, 100),  
 labels=["S" "I" "C"],  
 xlabel="Time (days)",  
 ylabel="Fraction of population")



## Searching for the optimal intervention constrained by maximum vaccination rate υ\_max, and cost

Parameters

p2 = copy(p)  
p2[3] = 0.05; #Set maximum vaccination rate to 0.05  
β = p2[1]  
γ = p2[2]  
υ\_max = p2[3]  
υ\_total = 1 # Maximum cost  
  
S0 = u0[1]  
I0 = u0[2]  
C0 = u0[3]  
  
T = Int(tf/dt)  
  
silent = true;

Model setup

model = Model(Ipopt.Optimizer)  
set\_optimizer\_attribute(model, "max\_iter", 1000)  
if !silent  
 set\_optimizer\_attribute(model, "output\_file", "JuMP\_ftc.txt")  
 set\_optimizer\_attribute(model, "print\_timing\_statistics", "yes")  
end;

Variables:

From their definition, the variables S, I and C are constrained to values between 0 and 1. We constrain our vaccination policy, υ(t) to lie between 0 and υ\_max.

@variable(model, 0 <= S[1:(T+1)] <= 1)  
@variable(model, 0 <= I[1:(T+1)] <= 1)  
@variable(model, 0 <= C[1:(T+1)] <= 1)  
@variable(model, 0 <= υ[1:(T+1)] <= υ\_max);

We discretise the SIR model using a simple Euler discretisation:

@expressions(model, begin  
 infection[t in 1:T], β \* I[t] \* dt \* S[t]   
 recovery[t in 1:T], γ \* dt \* I[t]   
 vaccination[t in 1:T], υ[t]\* dt \* S[t]   
 end);

We constrain the integral of the intervention to be less than or equal to υ\_total, assuming that the intervention is piecewise constant during each time step.

@constraints(model, begin  
 S[1]==S0  
 I[1]==I0  
 C[1]==C0  
 [t=1:T], S[t+1] == S[t] - infection[t] - vaccination[t]  
 [t=1:T], I[t+1] == I[t] + infection[t] - recovery[t]  
 [t=1:T], C[t+1] == C[t] + infection[t]  
 dt \* sum(υ[t]\*S[t] for t in 1:T+1) <= υ\_total  
end);

This scenario’s objective is to minimise the total number of cases:

@objective(model, Min, C[T+1]);

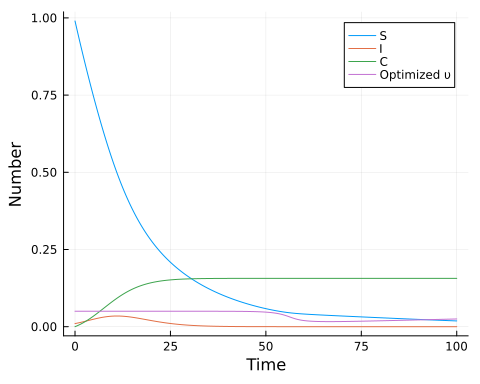
if silent  
 set\_silent(model)  
end  
optimize!(model)

termination\_status(model)

LOCALLY\_SOLVED::TerminationStatusCode = 4

S\_opt = value.(S)  
I\_opt = value.(I)  
C\_opt = value.(C)  
υ\_opt = value.(υ);

plot(ts, S\_opt, label="S", xlabel="Time", ylabel="Number")  
plot!(ts, I\_opt, label="I")  
plot!(ts, C\_opt, label="C")  
plot!(ts, υ\_opt, label="Optimized υ")



With the optimised vaccine intervention, we can observe that the maximum number of fraction of infected is

peak\_value\_opt, peak\_index\_opt = findmax(I\_opt)   
println("The maximum fraction of infected at a `dt` time is: ", peak\_value\_opt)

The maximum fraction of infected at a `dt` time is: 0.034552884095835804

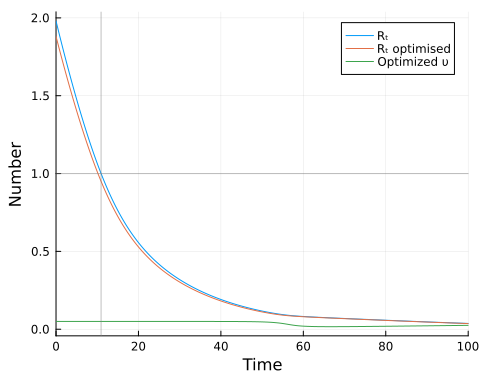
Again, we can calculate the effective reproductive number, Rₜ′ in the presence of the intervention:

Rₜ\_opt = β.\* S\_opt ./γ #Not taking into account the intervention  
Rₜ′\_opt = Rₜ\_opt .\* (1 .- υ\_opt); #Taking into account the intervention

And the time at which Rₜ==1 using a root-finding approach:

Rₜ\_interp = CubicSpline(Rₜ\_opt,ts)  
f(u, p) = [Rₜ\_interp(u[1]) - 1.0]  
u0 = [(tf-t0)/5]  
Rtprob = NonlinearProblem(f, u0)  
Rtsol = solve(Rtprob, NewtonRaphson(), abstol = 1e-9).u[1];

plot(ts, Rₜ\_opt, label="Rₜ", xlabel="Time", ylabel="Number", legend=:topright, xlim=(0,100))  
plot!(ts, Rₜ′\_opt, label="Rₜ optimised")  
plot!(ts, υ\_opt, label="Optimized υ")  
vline!([Rtsol], color=:gray, alpha=0.5, label=false)  
hline!([1.0], color=:gray, alpha=0.5, label=false)



## Discussion

Assuming that a vaccine is available at the start of an epidemic, the results in this example suggest that the optimal policy is to vaccinate early and at the maximum level available until the vaccine supply is exhausted.

The plot of Rₜ over time shows that, in this scenario as well as in the lockdown scenario, the optimal policy does not aim to keep Rₜ at or below 1 to prevent an increase in the infected population. Instead, it focuses on using the available vaccine supply to achieve the lowest possible total number of cases.