

Modelado Orientado a Objetos de Sistemas Poblacionales y Epidemiológicos en Tiempo Discreto

Camila Koatz¹, Mariana Bergonzi¹, and Ernesto Kofman¹

FCEIA-UNR, Pellegrini 250, Rosario, 2000, Santa Fe, Argentina
 CIFASIS-CONICET UNR, 27 de Febrero 210 bis, Rosario, 2000, Santa Fe, Argentina
 camilakoatz@gmail.com, bergonzi,kofman@cifasis-conicet.gov.ar

Abstract. Presentamos una librería básica orientada a objetos que permite representar modelos epidemiológicos compartimentales de tiempo discreto con retardos explícitos. Los modelos pueden diseñarse con distintas complejidades: poblaciones variables (con nacimientos y muertes) o constantes; puede incluir entrada de personas ajenas a la población inicial y también salidas. La librería también brinda la posibilidad de representar de manera muy simple epidemias transmitidas por un vector.

Además de la descripción de la librería, el presente trabajo muestra modelos de ejemplo y resultados simulación en un modelo de dengue que tiene en cuenta fenómenos meteorológicos que se ajustan a datos reales.

Keywords: Modelos Compartimentales, Epidemiología, Modelica

Object-Oriented Modeling of Discrete-Time Population and Epidemiological Systems

Abstract. We present a basic object-oriented library designed to represent discrete-time compartmental epidemiological models with explicit delays. The models can be built with varying levels of complexity: with either constant or variable populations (including births and deaths), and allowing for the inflow and outflow of individuals external to the initial population. The library also enables a straightforward representation of vector-borne epidemics.

In addition to the library description, this work includes example models and simulation results of a dengue outbreak model that incorporates meteorological phenomena, calibrated using real-world data.

Keywords: Compartmental models, Epidemiology, Modelica

1 Introducción

Los modelos compartimentales utilizados en dinámica poblacional y epidemiología se suelen formular mediante sistemas de ecuaciones diferenciales ordinarias (ODEs por *Ordinary Differential Equations*), lo que implica una evolución

continua de la variable en el tiempo. Ante la necesidad de incorporar retardos explícitos inherentes a tiempos de incubación o recuperación prolongados, estos sistemas de ODEs resultan en ecuaciones diferenciales con retardos. Si bien estas ecuaciones con retardo permiten capturar mejor los tiempos de permanencia en cada estado de una enfermedad, tienen como contrapartida la dificultad para su simulación y análisis.

Además, debemos tener en cuenta que en determinadas regiones, las bajas temperaturas invernales eliminan por completo la población de mosquitos y los brotes suelen reiniciarse anualmente a partir de casos importados que luego infectan a los primeros vectores nacidos en la nueva temporada cálida. Los modelos en tiempo continuo, al tender asintóticamente a cero, no logran replicar esta desaparición completa de la población, lo que muestra otra limitación de los modelos continuos para representar correctamente la dinámica poblacional presente en estas regiones.

Una alternativa a las ODEs y sus versiones con retardo, la brinda el uso de modelos en tiempo discreto (ecuaciones en diferencias). En este enfoque, los retardos pueden incorporarse de manera directa ya que las ecuaciones en diferencias con retardos (DDEs por Delay Difference Equations) son equivalentes a ecuaciones en diferencias regulares con un orden mayor. A diferencia de los modelos en tiempo continuo, los modelos en tiempo discreto permiten que una población se extinga completamente en ausencia de nacimientos durante un ciclo completo. Por lo que en estos casos, utilizar este enfoque permite representar de manera más realista la extinción total de poblaciones, un fenómeno crucial en el ciclo de vida de los mosquitos en regiones como la ciudad de Rosario.

No obstante, los modelos en tiempo discreto, especialmente aquellos que incorporan retardos, presentan una mayor complejidad en su formulación. Requieren la definición explícita de variables para cada día de vida de la población, así como las reglas que rigen su envejecimiento y movimiento entre días. Cuando se modelan enfermedades, esta complejidad se magnifica, ya que es necesario gestionar múltiples poblaciones (por ejemplo, mosquitos sanos e infectados, humanos susceptibles, infectados y recuperados), lo que dificulta la escritura de las ecuaciones del modelo.

Por estos motivos, decidimos desarrollar una librería orientada a objetos en tiempo discreto con retardos utilizando el lenguaje Modelica (ver Fritzson, 2015). Esta herramienta tiene como objetivo facilitar la construcción y simulación de modelos poblacionales y epidemiológicos en contextos donde el uso de tiempo discreto y dinámica con envejecimiento es necesario.

La idea básica de la librería es utilizar ciertas clases básicas para representar la dinámica interna de retardos de los compartimentos, a partir de las cuales se derivan clases especializadas para representar distintas poblaciones (huéspedes y vectores). Estos componentes permiten construir modelos poblacionales y epidemiológicos clásicos (SIR, SIRSI, SEIR, etc.) de manera muy simple, separando claramente la dinámica propia de todo modelo poblacional, de las reglas específicas de contagio y recuperación de cada enfermedad.

A modo de ejemplo, mostramos un modelo de evolución de una epidemia de Dengue y su ajuste con datos reales.

2 Conceptos Previos

2.1 Modelos Compartimentales

Existen muchos tipos de modelos matemáticos para representar, analizar y predecir los comportamientos y efectos producidos por una epidemia en una población determinada (ver Vynnycky and White, 2010). Los mismos pueden ser determinísticos o estocásticos, de tiempo continuo o discreto, de compartimientos u orientados a agentes, etc., dependiendo de las características que más nos interese modelizar.

Dentro de las distintas alternativas para describir modelos epidemiológicos, los modelos de compartimientos tipo SIR, SEIR y SEIRD se encuentran entre los más difundidos. Estos modelos en general se formulan en tiempo continuo (ecuaciones diferenciales) aunque hay variantes de tiempo discreto. El origen de estos modelos se remonta al trabajo de Kermack and McKendrick, 1927. Consiste en clasificar la población en tres grupos (o *compartimientos*) diferentes:

- $S(t)$: Población *susceptible* al virus en tiempo t . (Individuos que no están infectados, ni tienen inmunidad al virus por lo que pueden contagiar al entrar en contacto con un infectado).
- $I(t)$: Población *infectada* en tiempo t . (Individuos que están infectados y pueden contagiar a los susceptibles).
- $R(t)$: Población *recuperada* en tiempo t . (Individuos que ya han estado infectados y se recuperaron, adquiriendo inmunidad al virus, o bien murieron (por lo que también se la denomina población *removida* de la dinámica).

Cada individuo de la población estudiada pertenece a sólo uno de estos tres grupos y puede moverse de uno a otro. Una de las hipótesis es que los individuos recuperados adquieren inmunidad de por vida al virus, por lo que los movimientos permitidos pueden ser: Que un individuo de S pueda infectarse al estar en contacto con uno del grupo I , pasando a formar parte de este grupo. Uno del grupo I termina pasando al grupo R al recuperarse o morir. Entonces, el número de individuos en cada grupo cambia a través del tiempo. La cantidad total de individuos en la población se asume constante siendo $S(t) + I(t) + R(t) = N$

Existen otros modelos matemáticos basados en el modelo SIR, a los que se le agregan otros compartimientos o características adicionales: En el modelo SEIR se agrega a las ya mencionadas la categoría de:

- $E(t)$: Población *expuesta* en tiempo t . (Individuos que están incubando el virus, pero que todavía no tienen capacidad de contagio).

Por su parte en el modelo SEIRD se agrega la categoría D :

- $D(t)$: Población *muerta* en el tiempo t .

Es decir que se subdivide el compartimiento de individuos *removidos* entre los individuos *recuperados* (y que adquirieron inmunidad) de los *fallecidos* a causa del virus.

Modelos de tiempo continuo En poblaciones grandes las variables pueden ser tomadas como continuas y la relación entre los tres grupos se puede representar con un sistema de ecuaciones diferenciales ordinarias:

$$\frac{dS}{dt} = -\beta \cdot S \cdot I \quad (1a)$$

$$\frac{dI}{dt} = \beta \cdot S \cdot I - \gamma \cdot I \quad (1b)$$

$$\frac{dR}{dt} = \gamma \cdot I \quad (1c)$$

En donde β y γ son parámetros que representan las tasas de infección y de recuperación respectivamente. De la relación de estos parámetros surge un indicador fundamental: R_0 , el número de reproducción básico. Este factor indica a cuantas personas en promedio contagia un infectado durante su tránsito por la enfermedad.

Modelos en tiempo discreto Existen también modelos en los que se discretiza el tiempo, manteniendo la misma clasificación y dinámica del modelo SIR:

$$S_{n+1} = S_n \cdot (1 - \beta \cdot \Delta t \cdot I_n) \quad (2a)$$

$$I_{n+1} = I_n \cdot (1 + \beta \cdot \Delta t \cdot S_n - \gamma \cdot \Delta t) \quad (2b)$$

$$R_{n+1} = R_n + \gamma \cdot \Delta t \cdot I_n \quad (2c)$$

Modelos con retardos A los modelos mencionados, se les puede agregar retardos para modelar explícitamente los comportamientos que se manifiestan al cabo de un período de tiempo conocido. Un ejemplo puede ser el *tiempo de latencia* de una enfermedad, que es el tiempo desde que el individuo se expone al virus hasta que comienza a contagiar. También se puede mencionar el *tiempo de incubación*, que es el tiempo desde que una persona se expone al virus hasta que le aparecen los primeros síntomas.

Un ejemplo de modelos de tiempo discreto con retardo es el de COVID-19 de Bergonzi et al., 2020, que toma la siguiente representación:

$$N^E(t+1) = r_0 \cdot I(t) \cdot S(t)/N + u(t+1) \quad (3a)$$

$$S(t+1) = S(t) - r_0 \cdot I(t) \cdot S(t)/N \quad (3b)$$

$$E(t+1) = E(t) + N^E(t+1) - N^E(t+1 - \tau_I) \quad (3c)$$

$$I(t+1) = I(t) + N^E(t+1 - \tau_I) - N^E(t+1 - \tau_R) \quad (3d)$$

$$R(t+1) = R(t) + N^E(t+1 - \tau_R) \quad (3e)$$

$$(3f)$$

donde $N^E(t)$ es el número de individuos expuestos al virus en tiempo t , $U(t)$ es el número de casos importados en tiempo t , r_0 es el número de individuos que expone por día cada individuo infectado, y los restantes parámetros son los retardos que modelan los días que demora un individuo en comenzar a infectar (τ_I) y en recuperarse (τ_R).

La ventaja de este modelo que incorpora retardos, es que los mismos representan explícitamente los fenómenos característicos esenciales de la enfermedad. Al tener parámetros de interpretación directa, se facilita el estudio de diferentes escenarios mediante su modificación, lo cual sería más engorroso de representar en modelos clásicos de tiempo continuo. Por ejemplo, el efecto de una estrategia de seguimiento y aislamiento de contactos estrechos de casos confirmados (durante los N días previos a su detección) se puede capturar simplemente con una reducción del parámetro τ_R .

2.2 Modelica

Modelica (ver Fritzson, 2015) es un lenguaje orientado a objetos de modelado de sistemas dinámicos. El mismo permite definir clases y cuenta con herramientas de herencia, composición y conexiones para representar modelos complejos a partir de sub-modelos simples. La dinámica de las partes continuas de los modelos en las clases básicas se representa mediante ecuaciones diferenciales algebraicas acausales mientras la dinámica discreta se describe mediante algoritmos.

La naturaleza orientado a objetos y la descripción acausal de Modelica facilita la composición de modelos multi-dominio y la organización y reutilización de código en librerías. Un ejemplo es la *Modelica Standard Library*, un repositorio de componentes de distintos dominios de la física (eléctrico, mecánicos, hidráulico, térmicos, etc.) a partir del cual se pueden desarrollar nuevos modelos.

Estas ventajas y el hecho de que se trata de un lenguaje abierto, han transformado a Modelica en el lenguaje estándar moderno de modelado y simulación en muchos dominios, principalmente de la Ingeniería. Existen además distintas herramientas de modelado y simulación que utilizan este lenguaje tales como OpenModelica, Dymola, Wolfram SystemModeler, entre otras.

3 Librería Poblacional

La librería desarrollada es enteramente textual (no contiene capa gráfica) y está estructurada de manera que un paquete contiene las clases básicas que modelan la dinámica poblacional, otro paquete que contiene clases que, utilizando las clases básicas, representan modelos de huéspedes y de vectores de los que se derivan distintos tipos de modelos epidemiológicos clásicos (SIR, SIRSI, etc.).

3.1 Clases Básicas

El paquete define cuatro clases básicas, construidas mediante herencia entre ellas. Estas clases definen dinámicas generales de poblaciones en tiempo discreto sin brindar las leyes que rigen los nacimientos ni las migraciones.

La primera de estas clases, denominada `BasicPopulation` mostrada en la Lista 1.1 simplemente define el balance de población considerando nacimientos, muertes, ingresos y egresos. Estas variables son continuas lo que permite definir la interacción entre modelos a través de ecuaciones.

Lista 1.1: Clase básica para la evolución discreta de una población.

```
model BasicPopulation
  parameter Real n0=1e6 "Initial Population";
  discrete Real n(start=n0) "Population Size";
  Real arrivals "Arrivals per time unit";
  Real departures "Departures per time unit";
  Real births "Births per time unit";
  Real deaths "Deaths per time unit";
algorithm
  when sample(0,1) then
    n:=pre(n)+arrivals-departures+births-deaths;
  end when;
end BasicPopulation;
```

A partir de la clase `BasicPopulation`, utilizando herencia, se especializa la clase para el caso en que no hay nacimientos ni muertes (solo migración) dando lugar a la clase `MigrPop` de la Lista 1.2.

Lista 1.2: Clase básica para la evolución discreta de una población sin nacimientos ni muertes.

```
model MigrPop
  extends BasicPopulation;
equation
  births=0;
  deaths=0;
end MigrPop;
```

También heredando los atributos de la clase `BasicPopulation` se construye el modelo de una población estructurada en edad, en las cuales el número de individuos que nace va pasando por edades hasta un tiempo `Ltime`. Esta

clase, denominada `AgeingPop`, constituye la base para representar la dinámica de los compartimentos con tiempo de estadía finito (individuos infectados, por ejemplo, o vectores con vida corta).

Lista 1.3: Clase básica para la evolución discreta de una población transitoria.

```
model AgeingPop;
  extends BasicPopulation;
  constant Integer Ltime=10 "life time";
  discrete Real P[Ltime] "Population by age";
protected
  discrete Real dDeaths;
initial algorithm
  for k in 1:Ltime loop
    P[k]:=n0/Ltime;
  end for;
equation
  deaths=dDeaths;
algorithm
  when sample(0,1) then
    P[1]:=births;
    for k in 2:Ltime loop
      P[k]:=pre(P[k-1])+(arrivals-departures)/(Ltime-1);
    end for;
    dDeaths:=pre(P[Ltime]);
  end when;
end AgeingPop;
```

Notar que en esta clase el arreglo `P[k]` contiene el número de individuos de la población según el tiempo de permanencia en la misma. Además, se asume que la entrada y salida de individuos de la población es uniforme con la edad.

Finalmente, como especialización de esta última clase, se deriva la clase denominada `ClosedAgeingPop` que representa una población estructurada cerrada (sin migración).

Lista 1.4: Clase básica para la evolución discreta de una población transitoria cerrada.

```
model ClosedAgeingPop
  extends AgeingPop;
equation
  arrivals=0;
  departures=0;
end ClosedAgeingPop;
```

3.2 Modelos Epidemiológicos

El segundo paquete, denominado `Epidemics`, contiene diversas clases construidas a partir de las clases básicas antes descriptas. Estas nuevas clases no necesitan

describir la dinámica de las poblaciones, sólo establecer las relaciones entre las distintas poblaciones que incluyen.

La primera de estas clases define la base de un modelo SIR, que contiene tres poblaciones (**S**, **I**, y **R**), la primera y la última de tipo **MigrPop** (ya que la permanencia en dichos compartimentos se asume permanente) mientras que la población de individuos infectados **I** es de tipo **AgeingPop** ya que los individuos permanecen un tiempo **TR** en la misma. La Lista 1.5 muestra esta clase, denominada **BasicImpSIR**.

Lista 1.5: Clase básica para construcción de modelos SIR.

```
model BasicImpSIR
    Real n "Total population size";
    constant Integer TR=10 "Recovery time";
    parameter Real I0=1 "Individuals initially infected";
    parameter Real n0=1e6 "Initial population size";
    Basic.MigrPop S(n0=n0-I0),R(n0=0);
    Basic.AgeingPop I(n0=I0,Ltime=TR);
    Real dailyInfected "Individuals that get infected at the current time
        unit";
    Real importedInfected "Individuals that arrive infected at the
        current time unit";
equation
    n=S.n+I.n+R.n;
    S.arrivals=0;
    S.departures=I.births;
    I.births=dailyInfected;
    I.arrivals=importedInfected;
    I.departures=0;
    R.arrivals=I.deaths;
    R.departures=0;
end BasicImpSIR;
```

Notar que **I.births** y **I.deaths** no representan nacimientos y muertes, sino contagios y recuperaciones de individuos. La clase también considera la llegada de individuos infectados (casos importados). Vale aclarar además que esta clase básica no brinda las leyes de contagios (no define una ecuación para **dailyInfected**) ni define cuantos individuos infectados llegan (**importedInfected**).

Una especialización trivial de esta clase para el caso en que no hay casos importados la brinda la clase **BasicSIR** de la Lista 1.6.

Lista 1.6: Clase básica para construcción de modelos SIR sin casos importados.

```
model BasicSIR
    extends BasicImpSIR;
equation
    importedInfected=0;
end BasicSIR
```

Finalmente, la última clase básica del paquete la brinda el modelo de un vector, constituido por dos poblaciones: una susceptible **S** y otra infectada **I**.

A diferencia del caso SIR, ambas poblaciones son temporales ya que se considera que los vectores tienen un tiempo de vida corto independientemente de su condición de infectados o susceptibles. Esta clase, denominada **BasicVector** se describe en la Lista 1.7.

Lista 1.7: Clase básica para representación de vectores.

```
model BasicVector
    Real n "Total population";
    constant Integer Ltime=20 "Vector Lifetime";
    parameter Real n0=1e6 "Initial Vector population";
    parameter Real vt=0.4 "vertical transmission";
    parameter Real I0=1 "Initially infected vectors";
    Basic.AgeingPop S(Ltime=Ltime,n0=n0-I0);
    Basic.AgeingPop I(Ltime=Ltime,n0=I0);
    Real dailyInfected;
    Real dailyBirths;
equation
    n=S.n+I.n;
    I.births=dailyBirths*I.n/n*vt;
    S.births=dailyBirths-I.births;
    S.departures=dailyInfected;
    I.arrivals=S.departures;
    S.arrivals=0;
    I.departures=0;
end BasicVector;
```

De manera similar a la clase **BasicImpSIR**, esta clase no establece leyes para definir las nuevas infecciones (**dailyInfected**) ni nacimiento de nuevos vectores (**dailyBirths**). Sobre esto último sólo establece la proporción de nuevos nacimientos que resultan infectados o susceptibles en función del coeficiente de transmisión vertical **vt** (proporción de individuos que transmiten la infección a sus crías).

3.3 Modelos de Ejemplo

Utilizando las clases anteriores, se pueden crear de manera muy simple distintos modelos epidemiológicos. Por ejemplo, el sistema de la Lista 1.8 es un modelo SIR clásico en tiempo discreto donde el parámetro **R0** representa el número reproductivo básico (cuantos individuos son expuestos al virus por cada individuo infectado).

Lista 1.8: Modelo SIR

```
model SIR
    extends BasicSIR;
    parameter Real R0=2;
equation
    dailyInfected=R0/TR*I.n*S.n/n;
end SIR;
```

Notar que para la construcción de este modelo sólo fue necesario definir la ley para los nuevos infectados.

Otro ejemplo es el modelo de evolución de Dengue de la Lista 1.9. En el mismo, hay una población de vectores y otra de huéspedes. El modelo simplemente define el número diario de contagios de huéspedes y vectores y el número diario de nacimientos según una tasa de natalidad variante en el tiempo (lo que simula evoluciones climáticas).

Lista 1.9: Modelo simple de la evolución del dengue

```
model Dengue
    BasicVector V(n0=1);
    BasicSIR H(TR=5);
    parameter Real b=0.3 "Bites per day";
    parameter Real ph=0.75 "Probability of a host to get infected by a
        bite";
    parameter Real pv=0.75 "Probability of a vector to get infected by a
        bite";
    parameter Real bmax=0.098 "Maximum birth rate of vectors";
    Real br "Birth rate";
equation
    H.dailyInfected=H.S.n/H.n*V.I.n*b*ph;
    V.dailyInfected=H.I.n/H.n*V.S.n*b*pv;
    V.dailyBirths=V.n*br+1;
    br=bmax*(1+sin(time*6.28/365))/2;
end Dengue;
```

4 Resultados

Para ilustrar algunas posibilidades del uso de la librería mostramos a continuación los resultados de simular una extensión del modelo de dengue visto en la sección anterior, en la cual se tienen en cuenta datos meteorológicos reales de temperatura y lluvia, y se ajustan los parámetros para reproducir parcialmente los datos epidemiológicos de la ciudad de Rosario.

4.1 Modelo de Dengue con datos meteorológicos

Lista 1.10: Modelo de la evolución del dengue con datos meteorológicos

```
model ImpDengueTemp
    extends Utilities.Temperature;
    extends Utilities.Rain;
    BasicVector V(n0=1000);
    BasicImpSIR H(TR=5);
    parameter Real imported=2 "Infected individuals arriving per day";
    parameter Real b=0.3 "Bites per day";
    parameter Real ph=0.75 "Probability of a host to get infected by a
        bite";
```

```

parameter Real pv=0.75 "Probability of a vector to get infected by a
bite";
Real br "Birth rate";
equation
H.dailyInfected=H.S.n/H.n*V.I.n*b*ph;
V.dailyInfected=H.I.n/H.n*V.S.n*b*pv;
V.dailyBirths=V.n*br+1;
br=Utilities.BirthRateRain(temp,H.n,V.n,avgRain);
H.importedInfected=imported;
end ImpDengueTemp;

```

El modelo como puede verse, es esencialmente el mismo que de la Lista 1.9, pero cuenta con casos importados y además la tasa de natalidad se calcula en función de la temperatura, de las precipitaciones diarias y del tamaño de las poblaciones de huéspedes y vectores según la función `BirthRateRain`. Las clases `Temperature` y `Rain`, en tanto, brindan al modelo los datos meteorológicos diarios de los últimos 5 años almacenados en dos archivos.

La función mencionada para calcular la tasa de natalidad brinda una expresión que aumenta en torno a una temperatura óptima de 26 grados y de un nivel de precipitaciones óptimo de 10mm diarios. Además, dicha función disminuye a medida que aumenta la población de vectores respecto a los huéspedes (ya que se asume que el número de vectores no puede superar en mucho al número de huéspedes). La función está expresada en la Lista 1.11.

Lista 1.11: Función para la tasa de natalidad

```

function BirthRateRain
    input Real temp, nH, nV, avgRain;
    output Real br;
    protected
    parameter Real Topt=26,dT=7;
    parameter Real Ropt=8,dR=10;
    parameter Real bmax=0.6 "Maximum birth rate of vectors";
algorithm
    br:=bmax*max(0,1-abs(temp-Topt)/dT)*min(1,(nH/nV/5))*max(0,1-abs(
        avgRain-Ropt)/dR);
end BirthRateRain;

```

4.2 Resultados de simulación

Simulando el modelo para tres años de evolución con los parámetros indicados en la Lista 1.10 se obtienen los resultados de la Figura 1. En la misma se grafican los casos diarios (promediados semanalmente) y los casos simulados (multiplicados por un factor de detección de 0.06 ya que se estima que se detecta sólo una fracción de los casos reales). Si bien el objetivo del trabajo no era ajustar un modelo para reproducir los casos observados, el ajuste realizado corrobora en cierto modo la validez del modelo desarrollado.

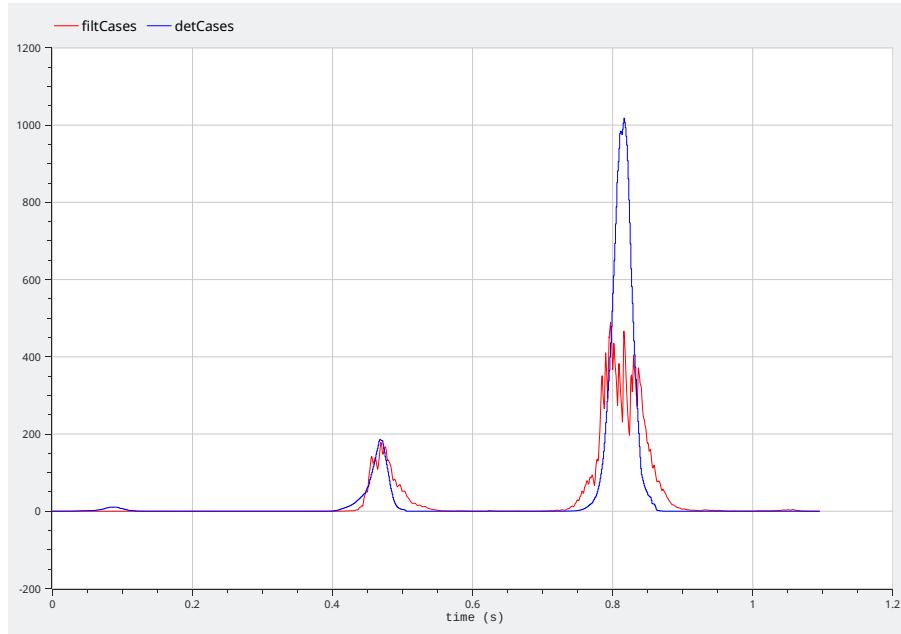


Fig. 1: Casos diarios reportados (rojo) y casos diarios simulados (azul)

Más allá de este detalle, la construcción del modelo de la Lista 1.10 termina resultando muy simple gracias al uso de la librería desarrollada. Otra cualidad de la técnica de modelado empleada es que los parámetros a ajustar se encuentran concentrados en una sola función (`BirthRateRain`) ya que los restantes parámetros son conocidos. Esto simplifica enormemente el trabajo.

5 Conclusiones y Trabajo Futuro

Se desarrolló una librería orientada a objetos de modelos poblacionales y epidemiológicos de tiempo discreto. Esta librería permite formular de manera sencilla modelos compartimentales de relativa complejidad.

La incorporación de retardos explícitos permitió una representación más precisa de las dinámicas poblacionales observadas en contextos climáticos particulares, en los que las bajas temperaturas generan períodos durante los cuales tanto la población de mosquitos como la de humanos infectados desaparecen por completo y el inicio de nuevos brotes se inician a partir de casos importados. Este tipo de comportamiento, no puede ser replicado con fidelidad mediante modelos en tiempo continuo, ya que en ellos las poblaciones tienden asintóticamente a cero, pero nunca se extinguen completamente.

Se mostró también el uso de la librería en distintos modelos epidemiológicos, incluyendo un modelo de evolución del dengue con el ajuste de parámetros a los datos reales de la ciudad de Rosario.

Este trabajo deja abierto a futuro el desarrollo de más clases y ejemplos para considerar casos más complejos incluyendo modelos SEIR, SEIRS, epidemias en múltiples poblaciones, entre otros casos.

References

- Bergonzi, M., Pecker-Marcosig, E., Kofman, E., & Castro, R. (2020). Discrete-time modeling of covid-19 propagation in argentina with explicit delays. *Computing in Science & Engineering*, 23(1), 35–45.
- Fritzson, P. (2015). *Principles of object-oriented modeling and simulation with modelica 3.3: A cyber-physical approach*. John Wiley & Sons.
- Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the royal society of london. Series A, Containing papers of a mathematical and physical character*, 115(772), 700–721.
- Vynnycky, E., & White, R. (2010). *An introduction to infectious disease modelling*. OUP oxford.