Simulations reproducibility: Inception using an epidemiological model

RED repo workshop

1 Context

The aim of this work is to study the reproducibility of simulations models.

Starting from the informal and formal description of a simple epidemiological system, we specify two different models of the same system. We use two different paradigms, the paradigm of dynamical systems and the agent or individual based paradigm. We implement the two models in various simulators and compare the results based on the same experimental plan. Simulation traces are compared two by two in order to build the following table of similarity indices (SI).

(Paradigm,Simulator)	(1,1)	(1,...)	(1,n)	(2,1)	(2,..)	(2,n)
(1,1)		*SI*	*SI*	*SI*	*SI*	*SI*
(1,...)			*SI*	*SI*	*SI*	*SI*
(1,n)				*SI*	*SI*	*SI*
(2,1)					*SI*	*SI*
(2,...)						*SI*
(2,n)						

We have the following paradigms and respective implementations:

- Dynamical systems:
  - (1,1): ODE solved with Tsitouras 5/4, free 4th order interpolant using the Julia mathematical software.
  - (1,2): ODE solve with QSS1 integrator using the irritator platform.
  - (1,3): ODE solve with QSS2 integrator using the irritator platform.
  - (1,3): ODE solve with QSS3 integrator using the irritator platform.
  - (1,4): ODE solve with QSS(?) integrator using the PyDEVS platform.
  - (1,5): ODE solve with solved with RK4 using the Gama platform.
  - (1,6): ODE solve with (?).
- Multi-Agent Systems:
  - (2,1): From scratch.

– (2,2): Implemented in the Gama platform.
– (2,2): Implemented in the Netlogo platform.
– (2,3): Implemented in the Cormas platform.

It is expected that different paradigms will be more dissimilar than different implementation of the same model on different simulators.

2 System description

We consider a simple epidemiological system. It is composed of individuals susceptible to a contagious disease. Infection occurs by close contacts (proximity) between susceptible and infectious individuals. A contact does not necessarily lead to the transmission of the disease. It depends on a transmission probability. Right after the infection, newly infected individuals enter an incubation period. At the end of the incubation period, infected individuals become and stay infectious for a certain duration. At the end of this infectious period, individuals recover and are immune to the disease, they can not be infected. After this immune period, they can get the disease again. The time resolution to follow the system's dynamic is the day.

Following this description, individuals can be in one of the following states: S for Susceptible, E for Exposed, I for infectious and R for Recovered, with $S \rightarrow E \rightarrow I \rightarrow R \rightarrow S$ the state transitions. The transitions $E \rightarrow I$, $I \rightarrow R$, and $R \rightarrow S$ depend on states lifespan as described in Table 1.

Name	Meaning	Associated State	Unit
t_e	Mean incubation period	E	*day*
t_i	Mean infectious period	I	*day*
t_r	Mean immune period	R	*day*

Table 1: States lifespan

The states lifespan varies between individuals. We assume the state lifespan value is a random number drawn from an exponential distribution with a shape parameter equal to t_e, t_i and t_r for the states E, I and R respectively.

The transition $S \rightarrow E$ depends on N_I, the number of infectious individuals in close contact with the susceptible one, on the duration Δt of the contact, and on the infection probability p. We can write:

$$p(\Delta t) = 1 - e^{-\beta N_I \Delta t}$$

or:

$$p(\Delta t) = \beta * N_I / N$$

with β the average transmission rate and N the number of possible contacts during Δt. N depends on the topology of contacts, i.e. the type of neighborhood on a 2D lattice for instance.

Name	Meaning	Unit
$\rho = 1/t_r$	The immunity loose rate	d^{-1}
β	The infectious rate	d^{-1}
$\sigma = 1/t_e$	The incubation rate	d^{-1}
$\gamma = 1/t_i$	The recovery rate	d^{-1}

Table 2: Parameters definition

3 Dynamical systems paradigm

3.1 Model specification with ordinary differential equations

We specify the system described in section 2 with a system of ordinary differential equations (ODEs). This lead to the classical continuous SEIRS model (equations 1) computing the proportion of individuals in each state at any $t > 0$. There is no explicit representation of space.

$$\begin{cases} \dot{S} &= \rho R - \beta(I)S \\ \dot{E} &= \beta(I)S - \sigma E \\ \dot{I} &= \sigma E - \gamma I \\ \dot{R} &= \gamma I - \rho R \end{cases} \tag{1}$$

3.1.1 State variables and parameters

- S - Susceptible, fraction of the population susceptible to infection.
- E - Exposed, fraction of the population infected but not yet contagious.
- I - Infected, fraction of the population infected and contagious.
- R - Recovered, fraction of the population recovered, i.e. no more contagious.

3.2 Experimental plan

Simulated time is two years (730 days). The initial conditions are: $S(0) = 0.999$, $E(0) = 0$, $I(0) = 0.001$ and $R(0) = 0$. The parameter values are given table 3.

Name	Value	Unit
ρ	1/365	d^{-1}
β	0.5	d^{-1}
σ	1/3	d^{-1}
γ	1/7	d^{-1}

Table 3: Parameters values

3.3 Simulation

3.3.1 Simulator

- Language: Julia (version 1.7.2)
- OS: Linux (x86_64-pc-linux-gnu)
- CPU: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz

3.3.2 Source code

We used the "DifferentialEquations" package [**?**] with an explicit Runge-Kutta method (Tsitouras 5/4, free 4th order interpolant). Below is the source code.

```
using DifferentialEquations
# Initial conditions
u0 = [0.999, 0.0, 0.001, 0.0]
# Parameters values
p = [1/365, 0.5, 1/3, 1/7]
tspan = (0.0, 730.0)
function seirs!(du,u,p,t)
  du[1] = p[1] * u[4] - p[2] * u[3] * u[1]
  du[2] = p[2] * u[3] * u[1] - p[3] * u[2]
  du[3] = p[3] * u[2] - p[4] * u[3]
  du[4] = p[4] * u[3] - p[1] * u[4]
end
prob = ODEProblem(seirs!,u0,tspan,p)
algo = Tsit5()
sol = solve(prob, algo, saveat = 0.01)
```

3.3.3 Results

Simulation results are stored in the "seirs-data.csv" file and illustrated by figure 3.3.3.

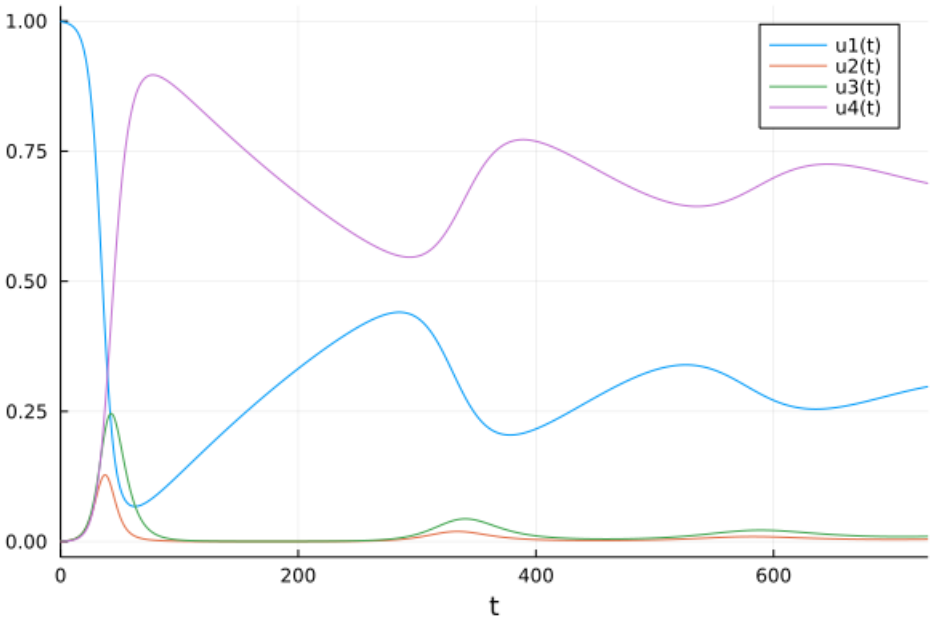

Figure 1: SEIRS model simulation with Julia.