



University
of Manitoba

Basic computational analysis of a mathematical model

02 – Using simulations

Julien Arino

University of Manitoba

julien.arino@umanitoba.ca

Fall 2024

The University of Manitoba campuses are located on original lands of Anishinaabeg, Ininew, Anisininew, Dakota and Dene peoples, and on the National Homeland of the Red River Métis.

We respect the Treaties that were made on these territories, we acknowledge the harms and mistakes of the past, and we dedicate ourselves to move forward in partnership with Indigenous communities in a spirit of Reconciliation and collaboration.

Using simulations ?

To caricature, suppose we have an IVP $x' = f(x, \mu)$, $x(t_0) = x_0$, where μ are parameters

We can find functional expressions telling us that, say, if $\Psi(\mu) < 0$, then the system has a certain behaviour and that this changes when $\Psi(\mu) > 0$

This type of functional relationship between model parameters and behaviour is what is in the first set of slides for this lecture

There are cases also where we need to numerically solve the IVP to obtain, say, the value of an equilibrium x^* , because there is no closed-form formula giving x^* as a function of μ

This type of work *uses simulations* and is what we are interested in here

Outline

Course description

Some toy epidemiological models

Using simulations of the ODE

Course description

Some toy epidemiological models

Using simulations of the ODE

• SIR model of an epidemic

Susceptible - Infected - Recovered

Infected - Susceptible

Recovered - Susceptible

Susceptible - Infected - Recovered

Infected - Susceptible

Recovered - Susceptible

Epidemic

Outbreak

Endemic

Sub-endemic

Extinct

Dead

Dead

Dead

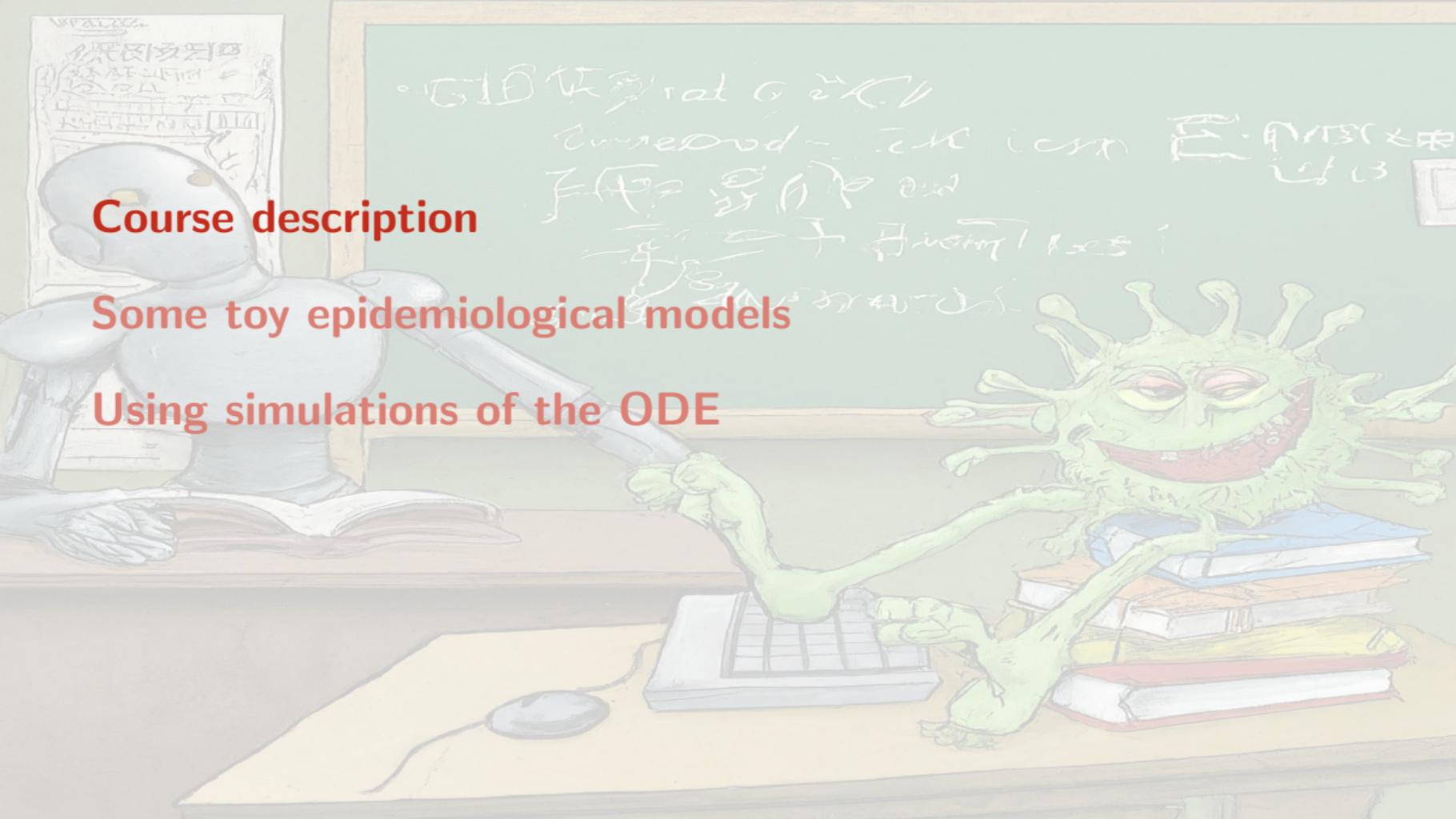
Dead

Dead

Dead

Dead

Dead



This is not a *vignette*!

The vignettes in this repo illustrate how to use R to consider several problems that a modeller is faced with

This is somewhat orthogonal: it takes the information in several of these vignettes and integrates it in the perspective of computational analysis

I am including it in this repo, however, because of the non-empty intersection between the two: this is R and is related to modelling

Course objectives

The objective of the course is to introduce notions used in the computational analysis of a mathematical model

This is not an exhaustive course on the subject, but rather an introduction to some basic concepts

See this as a minimal toolkit to get you started

If you are a graduate student of mine, take this as a hint: this is the type of stuff that I expect to see in your work

Note that I am not doing my job properly: I am skipping a very important part of any computational analysis by not doing a proper *return to biology*. This is an essential part of any computational analysis but is outside the scope of these slides

Course slides

These slides are produced using knitr in Rnw, i.e., R within \LaTeX , to illustrate some of the concepts

To generate them, you need to have R and \LaTeX installed on your computer and, preferably, RStudio

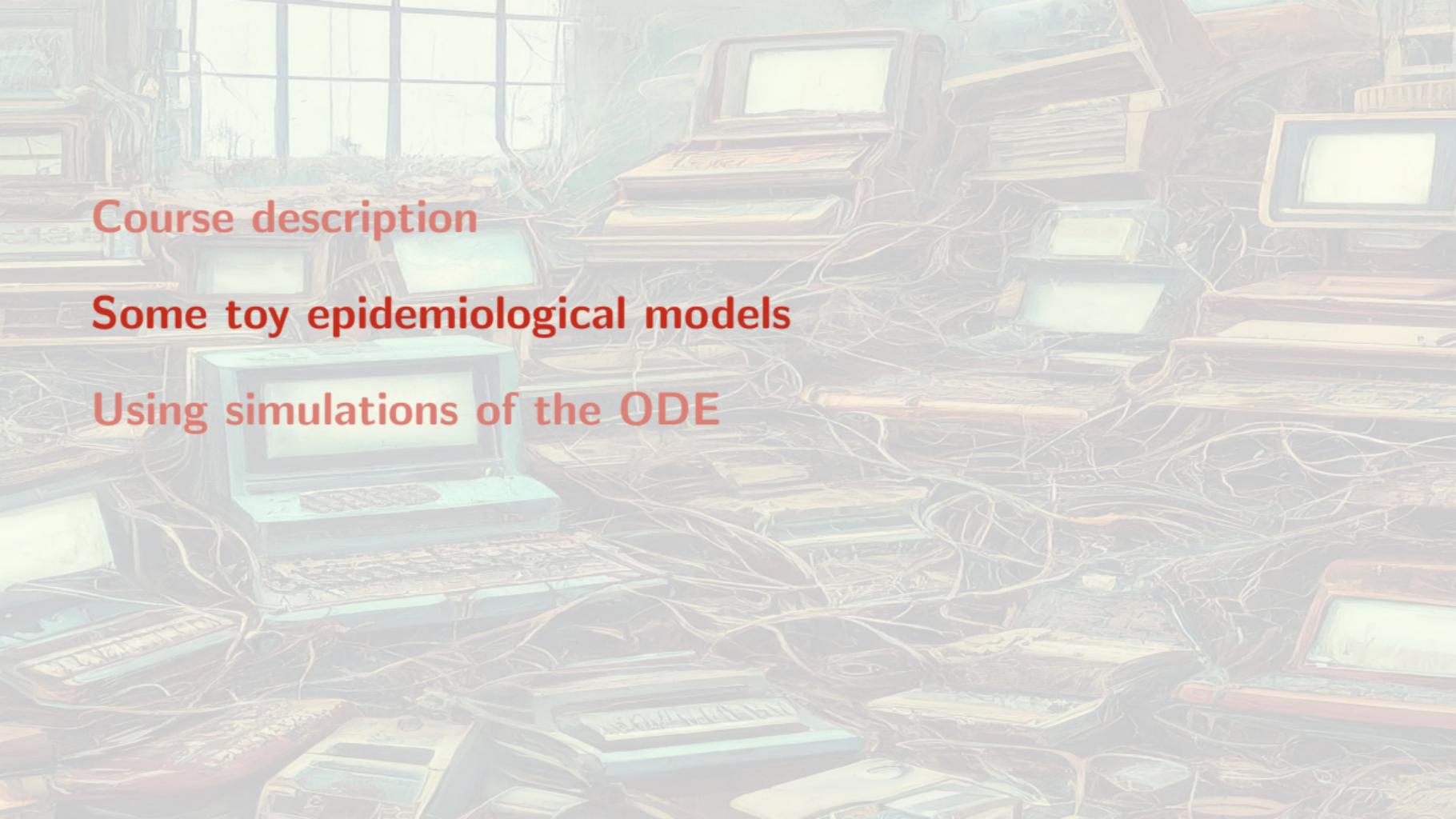
The code ensures that all the required packages are installed

Code chunks

In Rnw files, code chunks are delimited by <>>= and @

Code chunks are highlighted in the RStudio editor, so you should be able to identify them easily

I also generate an R file with all the code (`basic-computational-analysis.Rnw`). It is in the CODE directory of the repo. See the last slide for details

The background of the slide is a dense, abstract illustration. It depicts a multitude of vintage electronic devices, such as calculators, small personal computers, and monitors, all interconnected by a complex web of tangled wires. The colors are muted, with shades of brown, grey, and beige, giving it a somewhat aged and chaotic feel.

Course description

Some toy epidemiological models

Using simulations of the ODE

The toy models

I illustrate the methods using two toy models

Both are epidemiological models I have worked on

Some of the computational analysis is common to both models, some is specific

Some toy epidemiological models

The SLIAR epidemic model

The SLIARVS endemic model

Tackling the models computationally

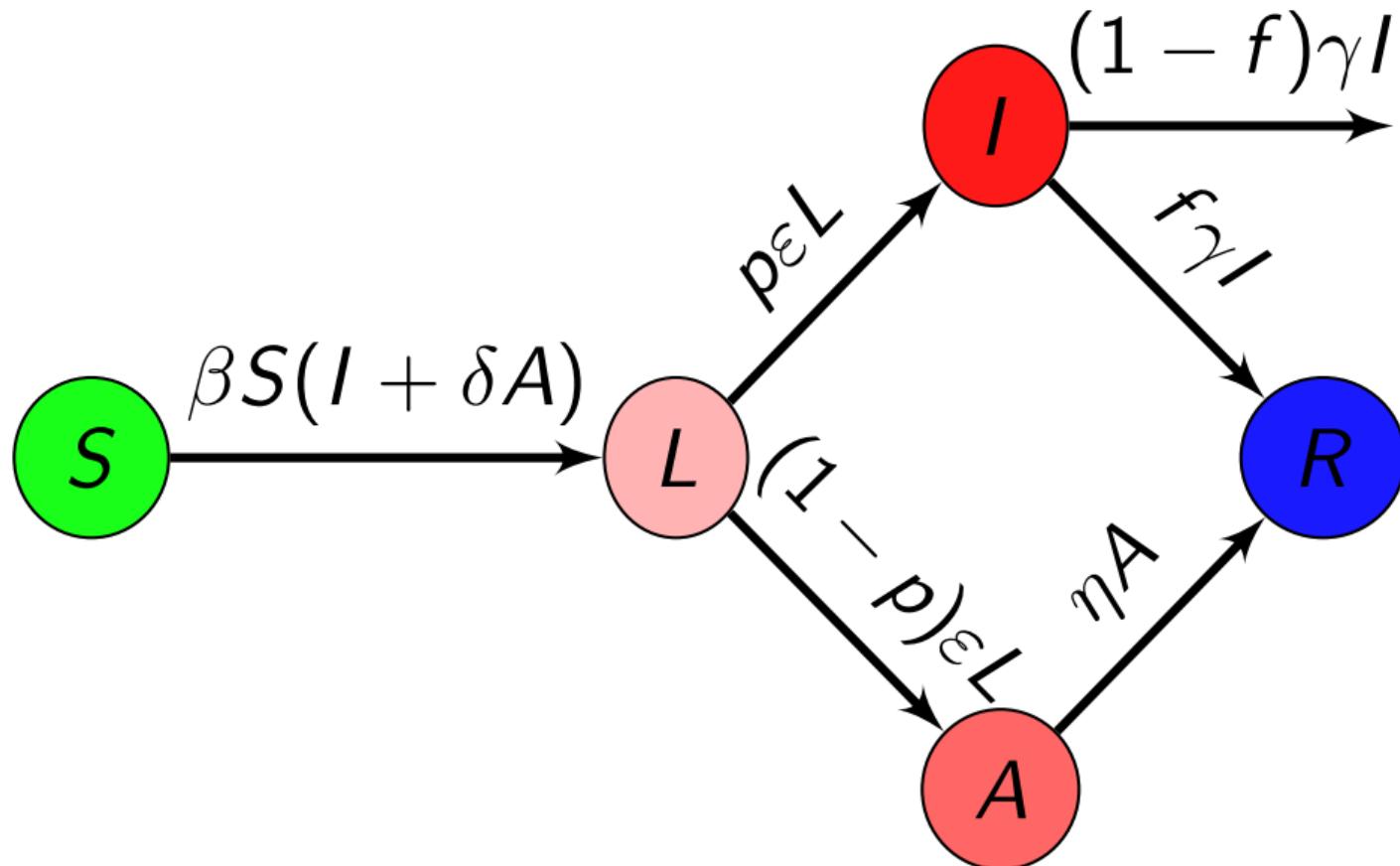
The SLIAR epidemic model

Kermack-McKendrick SIR (susceptible-infectious-removed) is a little too simple for many diseases:

- ▶ No incubation period
- ▶ A lot of infectious diseases (in particular respiratory) have mild and less mild forms depending on the patient

⇒ model with SIR but also L(atent) and (A)symptomatic individuals, in which I are now symptomatic individuals

The SLIAR epidemic model – Flow diagram



The SLIAR epidemic model – Equations

$$S' = -\beta S(I + \delta A) \quad (1a)$$

$$L' = \beta S(I + \delta A) - \varepsilon L \quad (1b)$$

$$I' = p\varepsilon L - \gamma I \quad (1c)$$

$$A' = p\varepsilon L - \eta A \quad (1d)$$

$$R' = f\gamma I + \eta A \quad (1e)$$

The SLIAR epidemic model – Behaviour

It's always a good idea to not barge into the computational analysis of a model without an understanding of its behaviour

This is an **epidemic** model: all its solutions go to a disease-free equilibrium

There is a **basic reproduction number** \mathcal{R}_0 (next slide) that determines whether the disease will spread or not. If $\mathcal{R}_0 < 1$, the disease dies out without first going through an outbreak if $\mathcal{R}_0 > 1$, the disease goes through an outbreak, then dies out

As with many epidemic models, we can also characterise the so-called **final size** of the epidemic

The SLIAR epidemic model – Basic reproduction number & Final size

We find the basic reproduction number

$$\mathcal{R}_0 = \beta \left(\frac{p}{\gamma} + \frac{\delta(1-p)}{\eta} \right) S_0 \quad (2)$$

The final size relation takes the form

$$S_0(\ln S_0 - \ln S_\infty) = \mathcal{R}_0(S_0 - S_\infty) + \frac{\mathcal{R}_0 I_0}{\rho} \quad (3)$$

with

$$\rho = \frac{p}{\gamma} + \frac{\delta(1-p)}{\eta}$$

Some toy epidemiological models

The SLIAR epidemic model

The SLIARVS endemic model

Tackling the models computationally

The SLIARVS endemic model

The SLIAR model is an epidemic model: all its solutions go to a disease-free equilibrium

Here we consider a complexification of the SLIAR epidemic model:

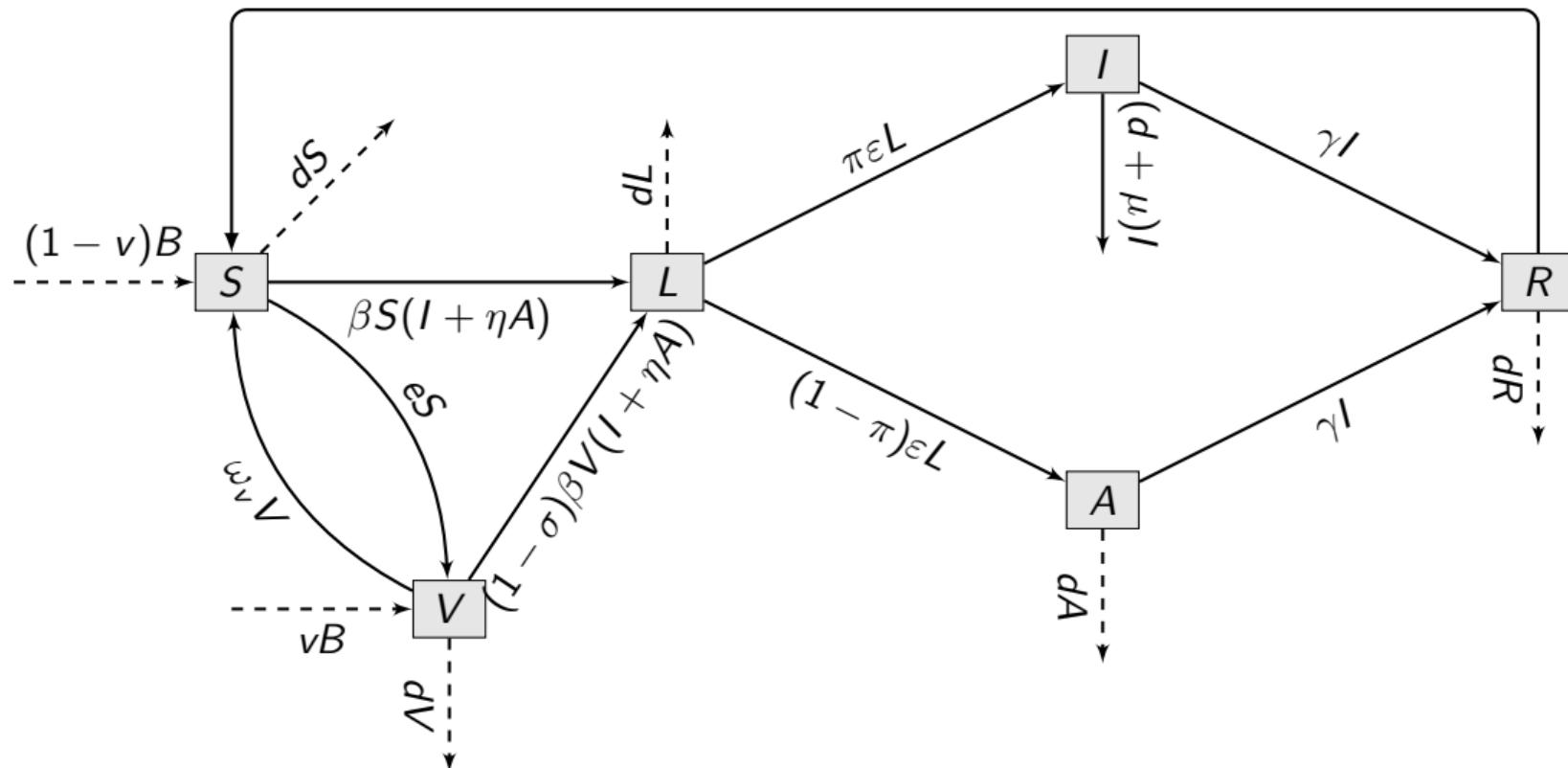
- ▶ Add vital dynamics (births and deaths), a.k.a. demography
- ▶ Add a vaccination compartment V , with imperfect and waning vaccine
- ▶ Interpret R as *recovered* (and immune) individuals instead of *removed*
- ▶ Add loss of immunity (waning immunity)

This makes the model **endemic**: it has an endemic equilibrium (EEP) and (roughly) \mathcal{R}_0 determines if the system goes to the DFE or the EEP

Arino & Milliken, Bistability in deterministic and stochastic SLIAR-type models with imperfect and waning vaccine protection, *Journal of Mathematical Biology* (2022)

The SLIARVS endemic model – Flow diagram

$\omega_r R$



The SLIARVS endemic model – Equations

$$S' = (1 - v)B + \omega_v V + \omega_r R - \beta S(I + \eta A) - (e + d)S \quad (4a)$$

$$V' = vB + eS - (1 - \sigma)\beta V(I + \eta A) - (\omega_v + d)V \quad (4b)$$

$$L' = \beta(S + (1 - \sigma)V)(I + \eta A) - (\varepsilon + d)L \quad (4c)$$

$$I' = \pi\varepsilon L - (\gamma + \mu + d)I \quad (4d)$$

$$A' = (1 - \pi)\varepsilon L - (\gamma + d)A \quad (4e)$$

$$R' = \gamma(A + I) - (\omega_r + d)R \quad (4f)$$

The SLIARVS endemic model – DFE

In (4) without equation for V' and with $v = e = \omega_v = 0$, disease-free equilibrium (DFE) has $\bar{S}_0 = B/d$

DFE of full (4) is $E_0 = (S_0, V_0, 0, 0, 0, 0)$, where

$$S_0 = \frac{(1-v)d + \omega_v}{e + \omega_v + d} \frac{B}{d} \quad \text{and} \quad V_0 = \frac{vd + e}{e + \omega_v + d} \frac{B}{d} \quad (5)$$

The SLIARVS endemic model – Reproduction numbers

With the combination parameter

$$\lambda = \beta\varepsilon \frac{(\gamma + \mu + d)\eta(1 - \pi) + \pi(\gamma + d)}{(\gamma + d)(\gamma + \mu + d)} \quad (6)$$

we have

$$\mathcal{R}_0 = \frac{\lambda}{\varepsilon + d} \bar{S}_0 \quad (7)$$

$$\mathcal{R}_v = \frac{\lambda}{\varepsilon + d} (S_0 + (1 - \sigma)V_0) \quad (8)$$

Some toy epidemiological models

The SLIAR epidemic model

The SLIARVS endemic model

Tackling the models computationally

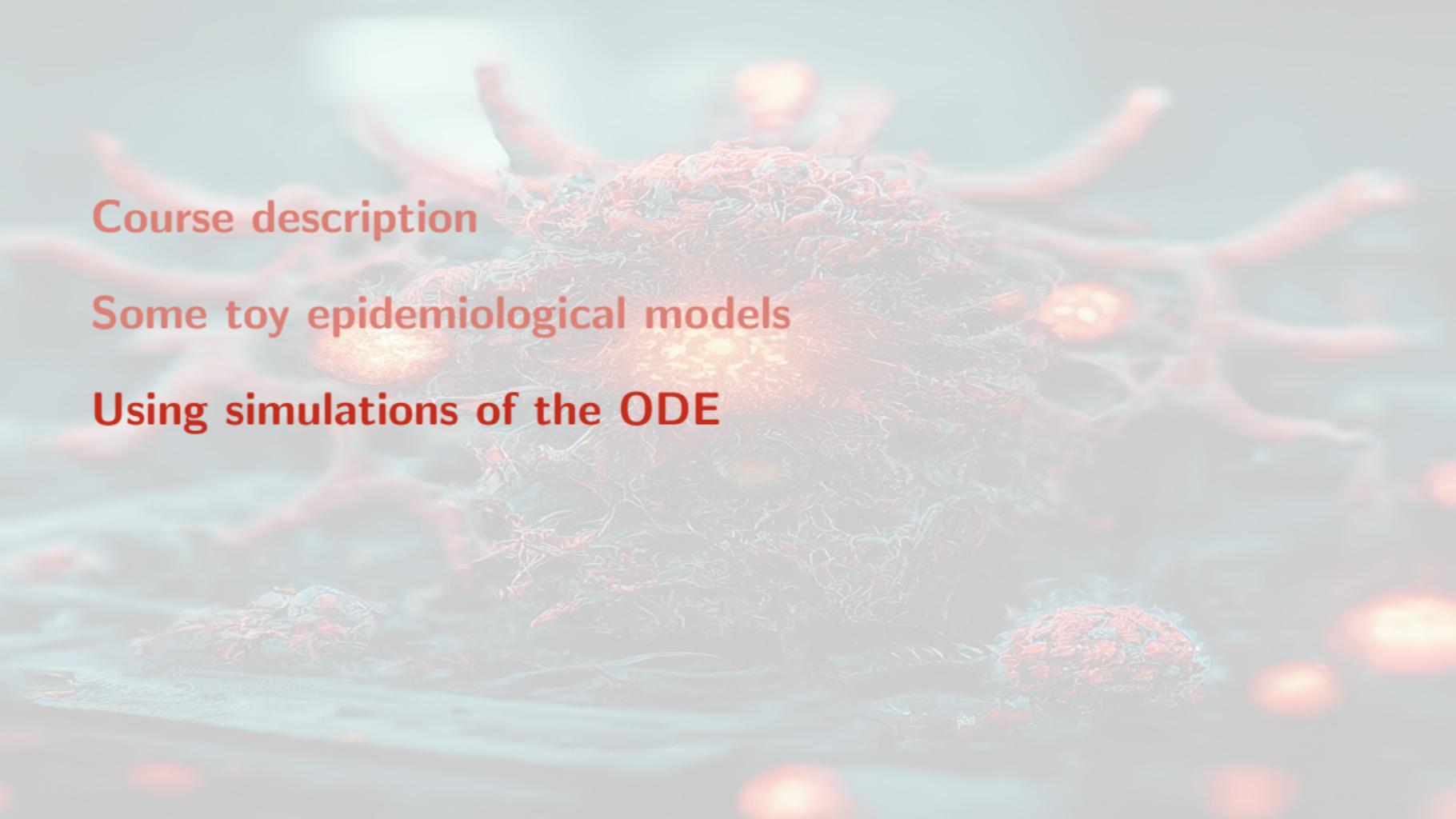
The models are systems of ODEs, we could simulate and show the result, but who cares if we just show the behaviour?

- ▶ System (1) goes to the DFE every time, after undergoing (or not) an epidemic depending on the value of \mathcal{R}_0
- ▶ System (4) goes to the DFE or the EEP, depending on the value of \mathcal{R}_0

Booooriiing!

We can still do things with the solutions, but we'll have to make it worthwhile...

To get more insight into the model, we can use the formula for the reproduction numbers (2), (7) and (8), the final size relation (3) and other quantities to study the model: this will show how these important quantities depend on parameters



Course description

Some toy epidemiological models

Using simulations of the ODE

In all we have done so far, ε and f have not been used

Indeed, they do not play a role in the computation of the basic reproduction number \mathcal{R}_0 given by (2) or the final size relation given by (3)

ε does play a role in determining the “speed” of the system, but we have not considered this aspect in our analysis so far

f helps determine how many individuals die of the disease and won’t be discussed here

CAN I HAVE THIS WRAPPED UP TO GO?

To finish, we use the command `purl` to generate an R file (`basic-computational-analysis.R`) in the `CODE` directory with all the code chunks in this Rnw file

```
# From https://stackoverflow.com/questions/36868287/purl-within-knit-duplicating-chunks
rmd_chunks_to_r_temp <- function(file){
  callr::r(function(file, temp){
    out_file = sprintf("../CODE/%s", gsub(".Rnw", ".R", file))
    knitr::purl(file, output = out_file, documentation = 1)
  }, args = list(file))
}
rmd_chunks_to_r_temp("basic-computational-analysis-2-simulations.Rnw")
## [1] "../CODE/basic-computational-analysis-2-simulations.R"
```

About that R file

Source the file `basic-computational-analysis-2-simulations.R` (in the CODE directory) in R to reproduce all the results in these slides

Some small changes are required; for instance, when sourcing (instead of knitting or interactively), `ggplot` figures are created but not printed, so in the R file, you need to print them “manually”

```
pp = ggplot(...)  
print(pp)
```