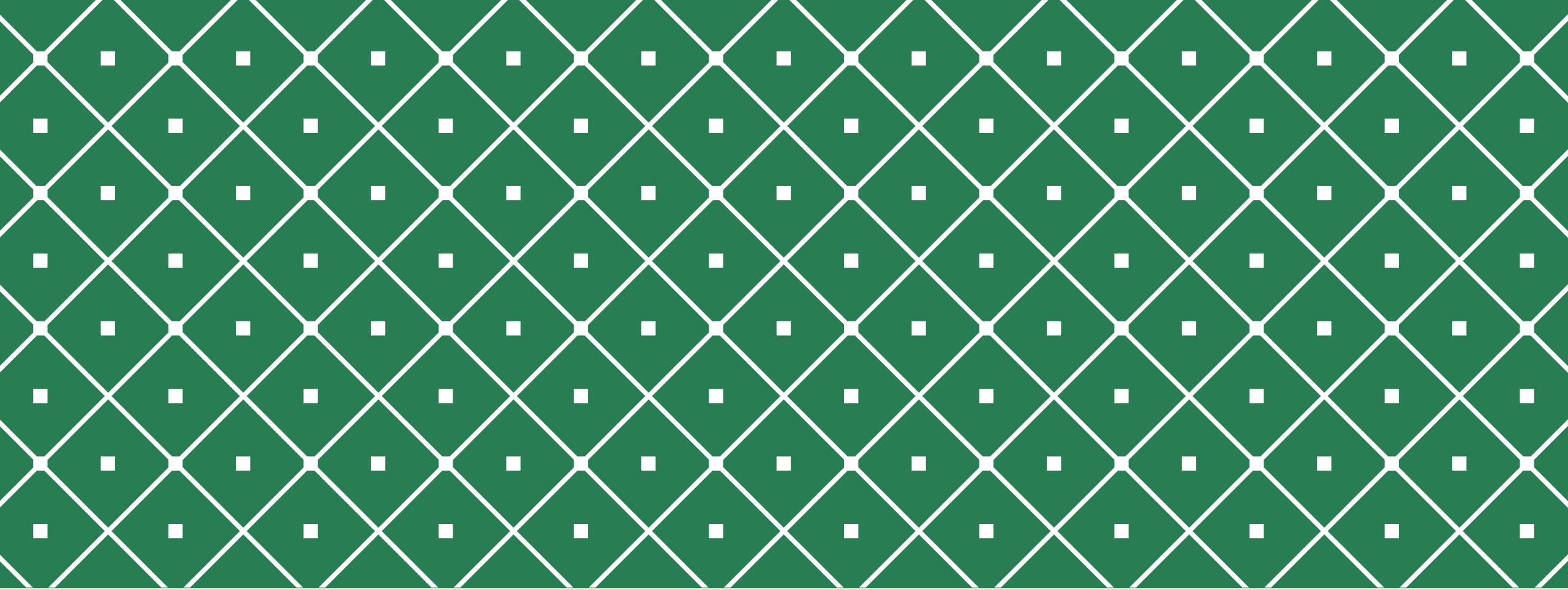
MODELLING INFECTIOUS DISEASES USING ~~GLOBAL STOCHASTIC~~ CELLULAR AUTOMATA

Mikler et al. 2005

BIO4134/8102

Martin Hanzel, Alanna Leale



HOW DOES DISEASE SPREAD DIFFER WITH SPATIAL STRUCTURE?

Classic SRI
Vs.
Simulation

SIMULATION VS. EQUATION MODELLING

PROS

- Incorporate spatial structure without complicated math
- Incorporate numerous variables/interactions (and track through time) with less math
- Enables stochasticity

CONS

- Predictions are difficult without actually running simulation
- Can't "see" how parameters are related and interact
- Too unrealistic or contrived (I beg to differ…)
- Stochasticity requires 1000+ runs → computing power!!!

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = +\beta SI - \gamma I$$

WHY DISEASE SIMULATION?

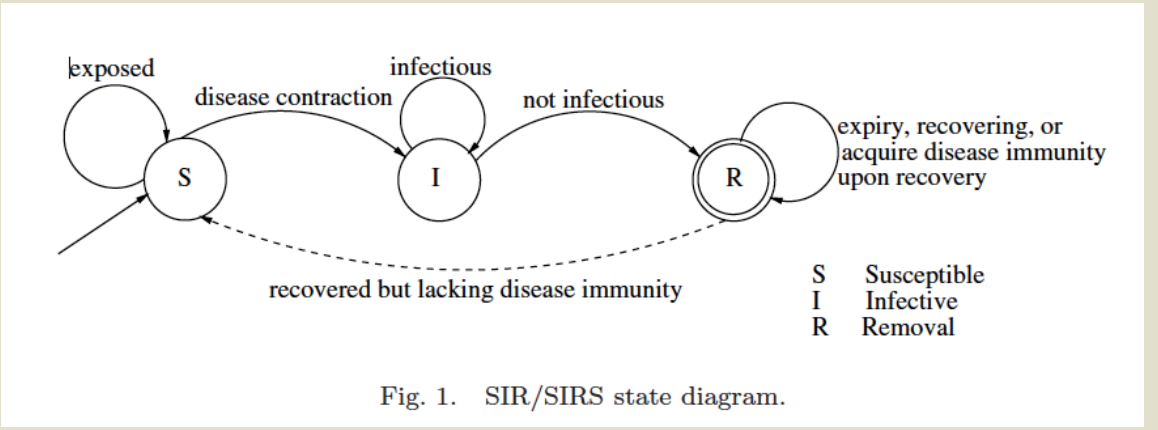


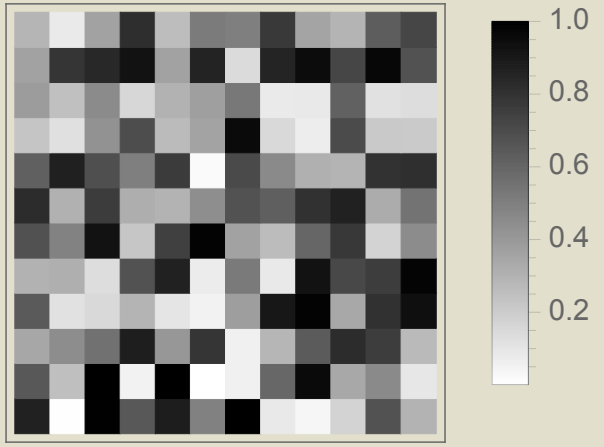
Fig. 1. SIR/SIRS state diagram.

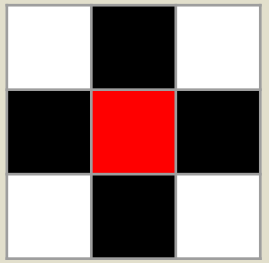
- Classic SIR models
  - Assumes homogenous mixing (individuals interact equally)
  - Spatial structure makes math too complicated!

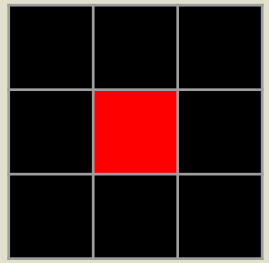
- Cellular automata / simulation
  - Interactions ~ geography, demography, environment…
  - Improve epidemiology models → better predict outbreaks

WHAT ARE CELLULAR AUTOMATA?

- Grid of cells
  - Each has distinct state ($S_{i,j}$) – continuous / discrete
  - Each has a neighbourhood of surrounding cells ($H_{i,j}$)







- Discrete time
  - $S_{i,j}(t+1) = S_{i,j}(t) \sim S_{neighb.}(t)$

$$s_{i,j}(t) = f(H_{i,j}(t-1)).$$

- Model *simulation*
  - Run one time step → calculate S for every cell
  - Repeat, repeat, repeat… (# iterations)

OUR MATHEMATICA MODEL

Initial States & Rules

- **SUSCEPTIBLE = 1** (if no infectious neighbours)
- **LATENT = 2** (1 time step)
- **INFECTIOUS = 3** (1 time step)
- **RECOVERED = 4** (permanent)

**you will add additional states (i.e., immune, latent1, latent2…)*
**you will change switch rules (i.e., return recovered to susceptible)*

IMPORTANT FUNCTIONS

Switch[*exp, form1 , value1 , form2, value 2…*]

- Determines "*value*" ($S_{i,j}(t+1)$) based on "*form*" ($S_{i,j}(t)$)

If[*condition, t, f*]

- Turns susceptible → infectious, based on neighbours

MemberQ[*list, form*]

- Does "*form*" ($S_{i,j}$=INFECTIOUS) exist in "*list*" of neighbour cells

***Think of it as "Contains[ ]"*

RandomReal[ ]

- Returns random value between 0 and 1(used for probability)

***Hint: Use for %immune, % susceptible etc.*

THINGS TO NOT WORRY ABOUT…

- DiseaseLibrary.wl file
- mRunModel

****DiseaseLibrary.wl must be in same folder as xxx.nb*

THINGS YOU MAY WANT TO KNOW…

- vnNeighbours & mooreNeighbours
  - Makes list of the state of every neighbour cell

- mCellNextState