

Agent-based models and the mathematical equations that describe them

Amy Hurford*, James Watmough, Joany Mariño, Anne McLeod, and Christina Prokopenko
*email: ahurford@mun.ca

Contents

1	Agent-Based Models	2
2	A non-spatial predator-prey model	2
2.1	The Gillespie Stochastic Simulation Algorithm	2
2.2	Code files	4
3	An ABM with local infection and reproduction on a lattice	7
3.1	A spatially explicit model	7
3.2	The pair approximation equations	9
3.3	Closing the pair equations	10
3.3.1	Ordinary Pair Approximation	10
3.3.2	Improved Pair Approximation	10
3.3.3	Singlet Density Approximation	11
3.4	Code files	11
4	Additional reading	13
A.1	Appendix	14
A.1.1	Software requirements	14
A.1.2	Background on software and package choices and installation problems	14
A.1.3	Debugging in NetLogo	15

1 Agent-Based Models

Agent-based models (ABMs) consider discrete individuals (agents), and are usually stochastic and spatially explicit. Formulating ABMs is intuitive for ecologists with a good knowledge of the intrinsic and extrinsic factors affecting their study organisms, although the value of simplifying assumptions and mathematical approximations can be underappreciated. Complex agent-based models are difficult to interpret and analyze: the stochastic ABM needs to be run many times and the results may consist of many variables, making it difficult to know which aspects to emphasize. When many variables interact, it can be difficult to build hypotheses as to which aspects of the organism's biology are useful to explain the model's results.

The goal of this workshop is to alleviate some of these challenges by showing how specifically formulated ABMs can be approximated by mathematical equations which are easier to comprehensively analyze. In Section 2, we explain how to formulate an ABM that is stochastic and with discrete individuals, so that it can be approximated with a system of ordinary differential equations. This first example is not spatially explicit and assumes that all individuals are equally likely to interact with one and other. In Section 3 we provide an example of a spatially explicit model where individuals are resident on a lattice and can interact with their nearest neighbours. This spatially explicit lattice-based ABM is approximated by a system of ordinary differential equations that describes the frequency of adjacent patches on the lattice with a particular status, for example, the frequency of adjacent patches that are both occupied by individuals of different epidemiological statuses. Section 4 describes additional reading. In the Appendix (Section A.1), we provide some explanations of the software choices and installation.

The software requirements to complete this workshop are described in Section A.1.1.

2 A non-spatial predator-prey model

2.1 The Gillespie Stochastic Simulation Algorithm

In this section, we consider a non-spatial ABM; however, despite the lack of space, this ABM is still more realistic than an ordinary differential equation, in that the ABM is stochastic and that individuals are discrete. We will use Gillespie's Stochastic Simulation Algorithm (Gillespie, 1977) to formulate our ABM. The ABM that we formulate will be approximated by the predator-prey equations with logistic birth rate for sheep and a type II functional response. Our ABM assumes only two species are present: the number of sheep (prey) is S , and the number of wolves (predators) is W . The model assumes the following events:

- E_1 : Each sheep will reproduce to generate, on average, $b(1 - S/K)$ new sheep per year, where K is the carrying capacity.
- E_2 : Each sheep will die of natural causes at an average rate of d_S deaths per year. This rate may be greater than 1, which would imply that, on average, sheep die of natural causes in less than a year.
- E_3 : Each sheep will die due to wolf predation at an average rate of $cW/(1 + cS)$ predation events per year.
- E_4 : Each wolf will reproduce in proportion to the rate of sheep predation, where the proportionality constant is ϵ , which is unitless, and should sensibly be less than 1 due to energy loss up the food chain.
- E_5 : Each wolf will die of natural causes at an average rate of d_W deaths per year.

It should be noted that the model does not track each wolf's sheep consumption and link individual-level sheep consumption to that wolf's reproduction. This individual-level bookkeep-

ing is unnecessary because the model is not spatial and all wolves are identical.¹ Across the wolf population there is a distribution of the number of births. Predation events are not assigned to a particular wolf, but if it is easier to conceptualize, it is possible to regard the wolves with higher than average sheep consumption also with having higher than average reproduction. This conceptualization is possible because all wolves are identical.

This description of the Gillespie algorithm is based on (Gillespie 1977; see also Section 5.1.1 Law of Mass Action in Muller and Kuttler 2015). We define the following notation:

μ	an index for each event
E_μ	description of the event indexed as μ
M	the total number of events described by the model
a_μ	the rate associated with the the μ^{th} event
α_0	$= \sum_{\mu=1}^M a_\mu$ i.e., the sum of all the rates α_μ
μ_i	a stochastic realization of the index for the event that occurred on the i^{th} iteration of the code.
τ_i	a stochastic realization of the time that the event μ_i occurred.

Therefore, for our predator-prey ABM as described above, in Gillespie's notation, $\mu \in \{1, 2, 3, 4, 5\}$, $M = 5$,

$$\vec{E} = \begin{bmatrix} \text{sheep birth} \\ \text{sheep natural mortality} \\ \text{sheep predation mortality} \\ \text{wolf birth} \\ \text{wolf natural mortality} \end{bmatrix}, \quad \text{and} \quad \vec{a} = \begin{bmatrix} bS \left(1 - \frac{S}{K}\right) \\ d_S S \\ \frac{cSW}{1+cS} \\ \frac{\epsilon cSW}{1+cS} \\ d_W W \end{bmatrix},$$

where the elements of the vectors are $\vec{E} = [E_1, E_2, E_3, E_4, E_5]^T$ and $\vec{a} = [a_1, a_2, a_3, a_4, a_5]^T$. The sum of the rates for all events is $a_0 = bS(1 - S/K) + d_S S + cSW/(1 + cS) + \epsilon cWS/(1 + cS) + d_W W$.

Following the reasoning outlined in Gillespie (1977), assuming that the time to the next event is exponentially distributed, then

$$\tau_i = \frac{1}{a_0} \log \left(\frac{1}{r_{1,i}} \right) \quad \text{where } r_{1,i} \sim U[0, 1], \quad (1)$$

where $r_{1,i} \sim U[0, 1]$ means that $r_{1,i}$ is random variable drawn from a uniform distribution defined over the interval from 0 to 1. The event that occurs is determined as,

$$\mu_i = \min_k \sum_{j=1}^k a_j > a_0 r_{2,i} \quad \text{where } r_{1,i} \sim U[0, 1], \quad (2)$$

where equation 2 is a re-written version of equation 21b from Gillespie (1977)). Equation 2 can be understood by considering a number line from 0 to a_0 that is divided up into subintervals with lengths corresponding to the rates of their associated events (recall that a_0 is the sum of all the rates); then, $a_0 r_{2,i}$ is a random point on the line used to determine the corresponding event. The cumulative sum (equation 2) defines the end of the subintervals for the event indexed by k .

¹Given that there are no individual-level variables, it could be argued that this example is not truly an agent-based model. None-the-less, this example is a good beginner exercise to help understand the algorithms that are used later for examples where each individual has associated individual-level characteristics.

The description of the event, E_{μ_i} , determines how the values of the state variables, $S(t)$ and $W(t)$ are updated. For example, if $\mu_i = 1$ then a sheep is born and the number of sheep increases by 1:

$$S(t + \tau_i) = S(t) + 1$$

The other state variable changes are:

$$\begin{aligned} S(t + \tau_i) &= S(t) - 1 && \text{for } \mu_i = 2, 3 \\ W(t + \tau_i) &= W(t) + 1 && \text{for } \mu_i = 4 \text{ and} \\ W(t - \tau_i) &= W(t) - 1 && \text{for } \mu_i = 5. \end{aligned}$$

Implemented in this way the ABM is approximated by the predator-prey equations,

$$\frac{dS}{dt} = bS \left(1 - \frac{S}{K}\right) - d_S S - \frac{cSW}{1 + cS}, \quad (3)$$

$$\frac{dW}{dt} = \frac{\epsilon cSW}{1 + cS} - d_W W. \quad (4)$$

These equations assume a type II functional response with handling time set to 1.

2.2 Code files

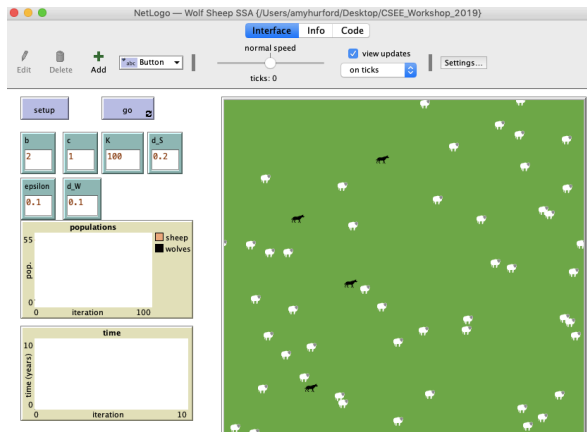
Wolf Sheep SSA.nlogo A Netlogo program that implements the Gillespie algorithm as described above.

Wolf_Sheep_Analysis.R An R script that imports the results of **Wolf Sheep SSA.nlogo** and numerically solves the system of ordinary differential equations (equations 3-4).

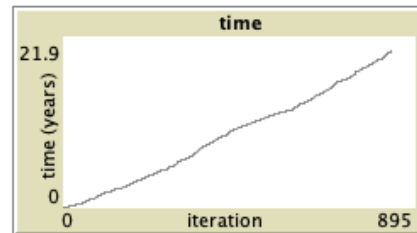
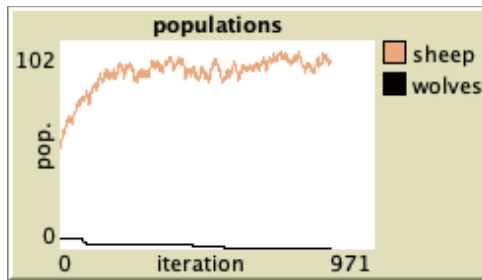
RosMac2019-08-13.R An R script that makes figures by calling to **Rosenzweig-MacArthurSSA.R**. **Rosenzweig-MacArthurSSA.R** uses the package **GillespieSSA** to simulate the stochastic process and has other precoded models available.

Instructions

1. Open **Wolf Sheep SSA.nlogo** and click the purple **setup** button.



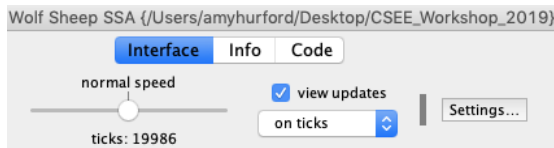
2. Click the purple **go** button. The simulation should begin running. Click **go** again to stop it. The plots track: (1) the number of sheep and wolves (y-axis) and the iteration of the code (x-axis), and (2) time (y-axis) and the iteration of the code (x-axis).



3. You can change the parameters and re-run the simulation.

b 2	c 1	K 100	d_S 0.2
epsilon 0.1	d_W 0.1		

4. Now consider the top middle/right of your Netlogo screen. You have the option to uncheck 'view updates': this will speed up the results of your simulation. There are also three options: 'Interface', 'Info', and 'Code'. Explore these different views.



5. In the 'Code' view you can identify the lines that end the code:

```
to go
; record the number of sheep and wolves
set nsheep count sheep
set nwolves count wolves
; several stop conditions
if nwolves + nsheep = 0 [ stop ]
if nwolves + nsheep > 10000 [ stop ]
if time-events > 20 [stop]
```

6. To create the data that we will analyze in R, we will need to run simulations several times and export the output. To do this we select 'Tools' and 'BehaviorSpace'. Select 'New' or 'Edit' from the window that opens. You need to make 3 changes to replace the default settings (left) with the desired settings (right):

Repetitions 1
run each combination this many times
☐ Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3
Measure runs using these reporters:
count turtles
one reporter per line; you may not split a reporter across multiple lines
☐ Measure runs at every step

Repetitions 10
run each combination this many times
☐ Run combinations in sequential order
For example, having ["var" 1 2 3] with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3
Measure runs using these reporters:
count sheep
count wolves
time-events
one reporter per line; you may not split a reporter across multiple lines
☒ Measure runs at every step
if unchecked, runs are measured only when they are over

7. Click 'Ok' and 'Run'. Then answer the following dialog boxes as follows:

Run options
☐ Spreadsheet output
☒ Table output
☐ Update view
☐ Update plots and monitors
Simultaneous runs in parallel 4
If more than one, some runs happen invisibly in the background.
Defaults to one per processor core.
Cancel OK

Save As: Wolf_Sheep_Output.csv
Tags:
Where: CSEE_Workshop_2019
Cancel Save

8. Now you can run your R script: `Wolf_Sheep_Analysis.R` (as long as you correctly set the path to find the output file). The output of the R script should resemble Figure 1.

```
30 ABMdata<-read.table("/Users/amyhurford/Desktop/CSEE_Workshop_2019/Wolf_Sheep_Output.csv",
```

9. In the R script, you can set `ABM_plot = 0` to bypass the parts of the code that visualize the results of the ABM. Change the parameter values for equations 3 and 4 and note how quickly the results via numerical integration are generated. Compare this to the time taken to simulation the ABM (in NetLogo) for different parameter values.

```
86 # Below we change K to 200
87 b <- 2
88 K<-200
89 c <-1
90 d_S<-0.2
91 epsilon <- 0.1
92 d_W <- 0.1
```

10. Try steps 3 - 8 for different parameter values.

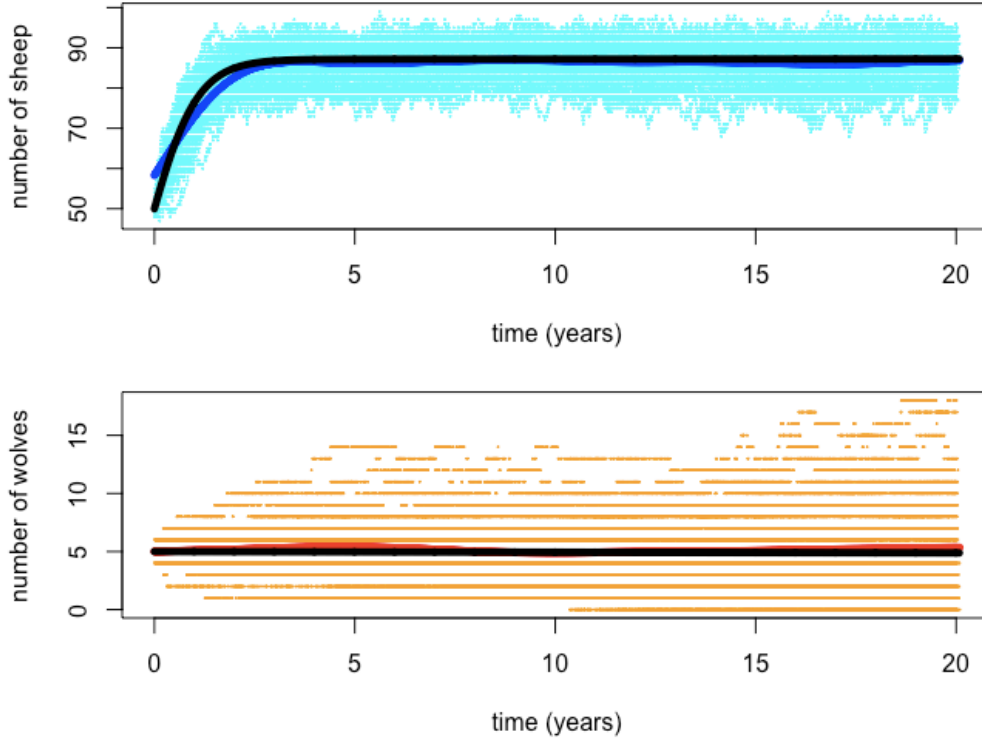


Figure 1: One hundred realizations of the ABM, where each realization is continued for 20 years and recording the the number of sheep (blue dots) and wolves (orange dots), with a spline fit shown as the red and blue lines. The solution to the equations 3 and 4 is the black solid line.

This example was selected, in part, because the mean of the ABM converges quickly to the numerical solutions of the ordinary differential equations 3 and 4. A similar example, but using the Lotka-Volterra predator-prey equations (exponential prey birth rate and a type I functional response) converged less well.

In addition to gains in computational time, an advantage of working with equations 3 and 4 is that analyses, such as those described at Wikipedia (2019), provide insight into parameter regions where stable equilibrium or cyclic dynamics occur.

The `GillespieSSA` package for R provides many tools to perform Gillespie SSA simulations. For an example using this package to simulate the Wolf-Sheep model described by equations 3-4, see the files `Rosenzweig-MacArthurSSA.R` and `RosMac2019-08-13.R`.

3 An ABM with local infection and reproduction on a lattice

3.1 A spatially explicit model

The previous example showed us that the mean of multiple realizations of a stochastic non-spatial ABM with discrete individuals implemented following the Gillespie algorithm can be approximated by the numerical solutions of an ordinary differential equation that represents the density of wolves and sheep as a continuous quantity. In this next example, we consider discrete sites spaced at regular intervals (a lattice). The model makes the following assumptions:

- The number of sites surrounding a focal site is z , i.e. $z = 2$ for one dimensional space and $z = 4$ for two dimensional space.

- Sites are either unoccupied, 0, or occupied by individuals susceptible, S , or infected, I with a disease.
- The infections dynamics are SI: infected individuals never recover.
- Local transmission occurs when the infection is transmitted to a susceptible individual from an infected nearest neighbour.
- Only susceptible individuals reproduce, reproduction can only occur if there is an unoccupied neighbouring site, and the offspring then occupies a neighbouring site. Offspring are born susceptible.
- The model does not explicitly describe movement.

The events that can occur are:

Events for Susceptibles:

E_1 : Natural mortality: Each susceptible dies of natural cases at an average per capita rate of d deaths per year.

E_2 : Reproduction: Each susceptible occupying a site with an adjacent empty site produces one offspring at a rate r per year.

E_3 : Local infection: Each susceptible individual, becomes infected from an infected neighbour at a rate β .

Events for Infecteds:

E_1 : Each infected dies of natural cases at an average rate of d deaths per year.

E_2 Each infected dies of disease-induced mortality at a rate of α deaths per year.

The ABM is coded in **Netlogo** following the Gillespie algorithm but with slight modifications to the definition of μ_i , the event that occurs on the i^{th} iteration of the code. Unlike the previous example (Section 2), where rates were equal for all prey (and all predators), for this model the neighbours of a focal individual will also affect the rates of the events. Let,

- $c_{j,m}$ be the rate that event E_j occurs for an individual indexed as m ,
- $b_m = \sum_j c_{j,m}$ be sum of all the rates for the individual m , and redefine
- $a_0 = \sum_m b_m$ as the sum of all the rates across all individuals.

For the spatially explicit ABM, we use an alternative version of equation 2 whereby on the i^{th} iteration of the code we: (1) randomly select an individual, l_i , such that,

$$l = l_i = \min_k \sum_{m=1}^k b_m > a_0 r_{2,i} \quad \text{where } r_{2,i} \sim U[0, 1], \quad (5)$$

where the probability of selecting any individual is proportional to the sum of their rates; and then (2) for the randomly selected individual, l , we randomly select an event,

$$\mu_i = \min_k \sum_{j=1}^k c_{j,l} > b_l r_{3,i} \quad \text{where } r_{3,i} \sim U[0, 1]. \quad (6)$$

The time to the next event, τ_i , is calculated using equation (1), but where a_0 is as defined above in this section as the third bullet point. We calculated μ_i using this two-step process to take advantage of **NetLogo**'s strength in cataloguing individual-level (**turtles-own**) properties, which were used to evaluate 5. If this same calculation were performed in **MATLAB** or **R**, which

have more functions available to work with vectors, we might have implemented equations 5 and 6 differently.

3.2 The pair approximation equations

For our spatially explicit disease model, we will derive a system of ordinary differential equations describing the change in the proportion of site pairs, $p_{\sigma\sigma'}$, where σ and σ' are sites that are either empty, 0, or occupied by individuals that are susceptible, S , or infected, I . The equations are derived in Sato et al. (1994), but we opt for the notation of Boots and Sasaki (1999), which we find more accessible (note that we set $P = 0$ in equations 1-6 of Boots and Sasaki (1999) because we assume only local transmission for our models). The rate of change in the proportion of sites that are a particular type of pair depends on the status of the neighbouring sites. We use $q_{\sigma/\sigma'\sigma''}$ to denote the probability that a neighbouring site is type σ for a pair of sites (σ' and σ''). The $/$ notation in $q_{\sigma/\sigma'\sigma''}$ indicates a conditional probability, where the fixed condition is the type of paired site. For a two-dimensional lattice, any particular site will have four neighbours, $z = 4$, and $\theta = 1/z$ is the proportion of nearest neighbour sites that are any one site. The dynamics of the change in the proportion of each type of paired sites in the landscape is described by the equations:

$$\frac{dp_{SS}}{dt} = 2r(\theta + (1 - \theta)q_{S/0S})p_{S0} - (2d + 2\beta(1 - \theta)q_{I/SS})p_{SS}, \quad (7)$$

$$\frac{dp_{S0}}{dt} = r(1 - \theta)q_{S/00}p_{00} + dp_{SS} + (d + \alpha)p_{IS} - (d + r(\theta + (1 - \theta)q_{S/0S}) + \beta q_{I/0S})p_{S0}, \quad (8)$$

$$\frac{dp_{00}}{dt} = 2p_{S0} + 2(d + \alpha)p_{I0} - 2r(1 - \theta)q_{S/00}p_{00}, \quad (9)$$

$$\frac{dp_{II}}{dt} = 2\beta(\theta + (1 - \theta)q_{I/IS})p_{IS} - 2(d + \alpha)p_{II}, \quad (10)$$

$$\frac{dp_{IS}}{dt} = r(1 - \theta)q_{S/0I}p_{I0} + \beta q_{I/SS}p_{SS} - (2d + \alpha + \beta(\theta + (1 - \theta)q_{I/SI}))p_{IS}, \quad (11)$$

$$\frac{dp_{I0}}{dt} = \beta(1 - \theta)q_{I/S0}p_{S0} + dp_{IS} + (d + \alpha)p_{II} - (d + \alpha + r(1 - \theta)q_{S/0I})p_{I0}, \quad (12)$$

Sato et al. (1994); Boots and Sasaki (1999). We note that the order of site types does not matter, such that the change in paired sites $S0$ and $0S$ are represented by equation 8, and the same is true of the other combinations of different site pairs: the ordering does not matter.

To understand equations 7-12, we will step through each term in equation 7, which describes the change in the proportion of paired sites that are both occupied by susceptible individuals. First, we note that an SS paired site is created from an $S0$ site, where reproduction occurs into the 0 site. The proportion of sites that are $S0$ is p_{S0} , and there are two possible orderings for this site type: $S0$ or $0S$. For these $S0$ paired sites, one of the nearest neighbours of the 0 site is the S site that comprises the other half of the pair, which is the fraction θ of all the neighbours. To calculate the status of the other $z - 1$ neighbours, we note that $q_{S/0S}$ is the probability that an $0S$ paired site has an S neighbour. The reproductive rate is r , and so the first term of the equation 7 describes the rate that new SS sites are created from $S0$ sites through reproduction. The model assumes that in some small increment of time only one change can happen, and so it is not possible that a 00 paired site can change to an SS paired site without first becoming an $S0$ paired site.

The last term in equation 7 describes the rate of loss of SS paired sites to $S0$ or SI sites. Natural mortality occurs at a rate d , and the rate that natural mortality occurs on either of the two SS sites is $2d$. In addition, infection may occur on either of the two sites at rate β , if a neighbouring site is I . Since the pair is SS , we know that each S site has 1 nearest neighbour that is not I . The fraction of remaining nearest neighbours is $1 - \theta$, and the probability that

one of these neighbours is infected, is $q_{I/SS}$. The remaining equations (8-12) can be interpreted similarly by considering the processes that create or destroy particular site pairs.

The frequency of each site type in the landscape is calculated as the sum,

$$p'_\sigma = \sum_{\sigma \in \{0, S, I\}} p_{\sigma\sigma'}. \quad (13)$$

Note that SS site pairs do not contribute twice as many S individuals to the population as $S0$ patches because there are two ways to make a $S0$ pair: either $S0$ or $0S$.

3.3 Closing the pair equations

The challenging aspect of equations 7-12 is that the probabilities for the neighbouring sites of a pair, $q_{\sigma/\sigma'\sigma''}$, can only be accurately calculated if the proportion of specific site triples are known, which in turn, can only be accurately calculated if the proportion of site quadruples is known, and so forth. While we could derive equations for the changes in the number of site triples, or even higher combinations of site pairings, our goal is to approximate the ABM, rather than to create an alternative version of a complex model. The process of ending the infinite dependency of site combination changes on higher order site combinations through approximation is referred to as ‘closure’. The R script that we provide considers three types of closure: (1) the ordinary pair approximation (OPA), (2) the improved pair approximation (IPA, Sato et al. (1994)), which may be more accurate for the specific SI disease model that we are studying, and 3) the Singlet Density Approximation (SDA; Matsuda et al. (1992)).

3.3.1 Ordinary Pair Approximation

The OPA assumes that $q_{\sigma/\sigma'\sigma''} \approx q_{\sigma/\sigma'}$, which means that the approximation assumes that the status of the other patch, σ'' , does not affect the types of neighbours that surround the focal patch. The formula for conditional probabilities states that,

$$q_{\sigma/\sigma'} = \frac{p_{\sigma\sigma'}}{p_{\sigma'}}$$

where $p_{\sigma\sigma'}$ for all possible $\sigma\sigma'$ combinations can be calculated by numerically integrating equations 7-12 where $p_{\sigma'}$ is the proportion of sites that are of type σ' , which is determined by summing proportion of all site pairs that have at least one site of the type σ' . The notation $p_{\sigma\sigma'}$ does not attach any significance attached to the site type appearing in the first or second subscript position. Therefore, the OPA could equivalently be calculated as $q_{\sigma/\sigma'\sigma''} \approx q_{\sigma/\sigma'}$ or $\approx q_{\sigma/\sigma''}$.

3.3.2 Improved Pair Approximation

The IPA notes that similar types of sites tend to aggregate in space. Such that where $0 \leq \epsilon \leq 1$,

$$q_{S/00} = \epsilon q_{S/0}, \quad (14)$$

since the combination of two empty sites is less likely to have a neighbour that is S , than just one empty site whose other partner might well be S . Furthermore, the IPA assumes,

$$q_{I/S\sigma} \approx q_{I/S} \quad \text{and} \quad q_{S/0I} \approx q_{S/0}, \quad (15)$$

which are OPA-type approximations. Finally, equations 14 and 15 and the identity relationship for conditional probabilities, then implies that,

$$q_{S/0S} = 1 - q_{I/0} - \epsilon q_{0/0} \quad (16)$$

and no other approximations are needed since other possible $q_{\sigma/\sigma'\sigma''}$ do not appear in equations 7-12. Sato et al. (1994) estimates that for a two-dimensional lattice ($z = 4$), that $\epsilon = 0.8093$. This $\epsilon = 1$ then the IPA reduces to an OPA.

3.3.3 Singlet Density Approximation

The Singlet Density Approximation (SDA) approximates $q_{\sigma/\sigma'\sigma''}$ by assuming the probability that a σ site is a neighbour of a $\sigma'\sigma''$ pair is equal to the landscape-wide frequency of σ , i.e.

$$q_{\sigma/\sigma'\sigma''} = p_{\sigma}, \quad (17)$$

3.4 Code files

Local SI.nlogo A Netlogo program that performs a Gillespie algorithm implementation of the pair approximation equations for the local infection spread model (equations 7-12).

Local_SI_Analysis.R This R file numerically solves the pair equations 7-12 and is able to reproduce Figure 3 of Sato et al. (1994) (see Figure 2). The file also imports the results of Local SI.nlogo simulations that have been saved as the files *SI_Output.csv* and *SI_Output2.csv* and compares the agreement between the ABM and the numerically solved pair equations with four different closures (Figure 4) .

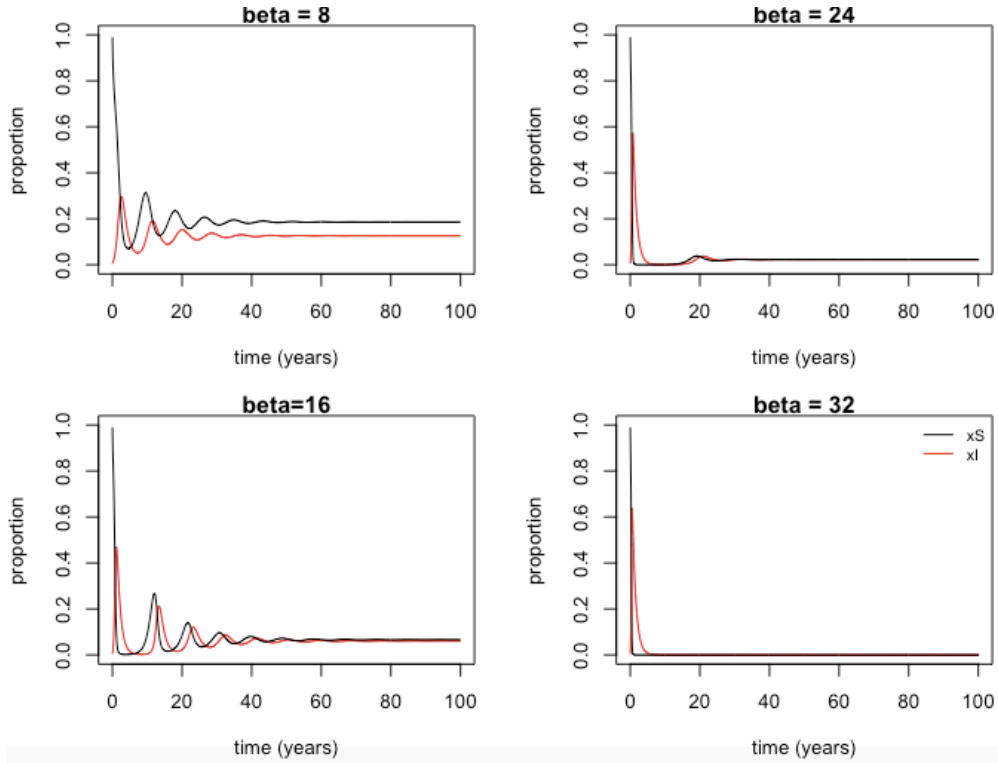


Figure 2: We numerically solve equations 7-12 using the IPA to reproduce Figure 3 of Sato et al. (1994). The parameter values are $r = 4$, $\alpha = 0$, $d = 1$, $\epsilon = 0.8093$ and β as described by the panel titles.

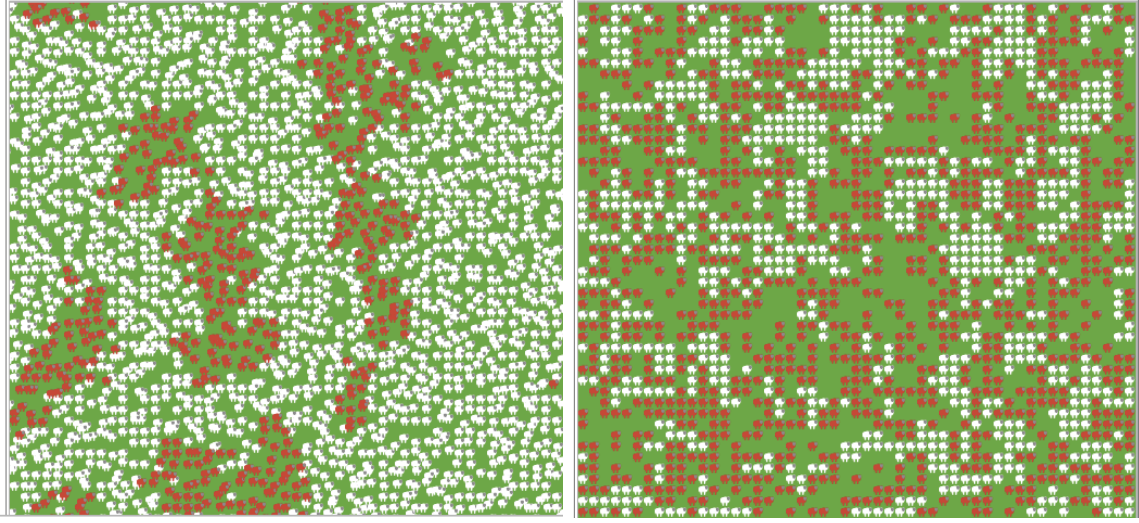


Figure 3: Simulation results generated by `Local SI.nlogo`. When the transmission rate is much higher than the natural mortality rate infections are found in clusters (left), but when the transmission rate is more similar to the natural mortality rate empty space frequently becomes available for susceptible individuals to reproduce into allowing susceptible sheep to persist in the population long term (right). For the panel on the left, $\beta = 4$, $t = 0.55$ years, and eventually all sheep become infected. For the panel on the right, $\beta = 1$ and the figure shows a dynamic steady state reached after 20 years. The other parameter values are $r = 4$, $d = 1$ and $\alpha = 0$.

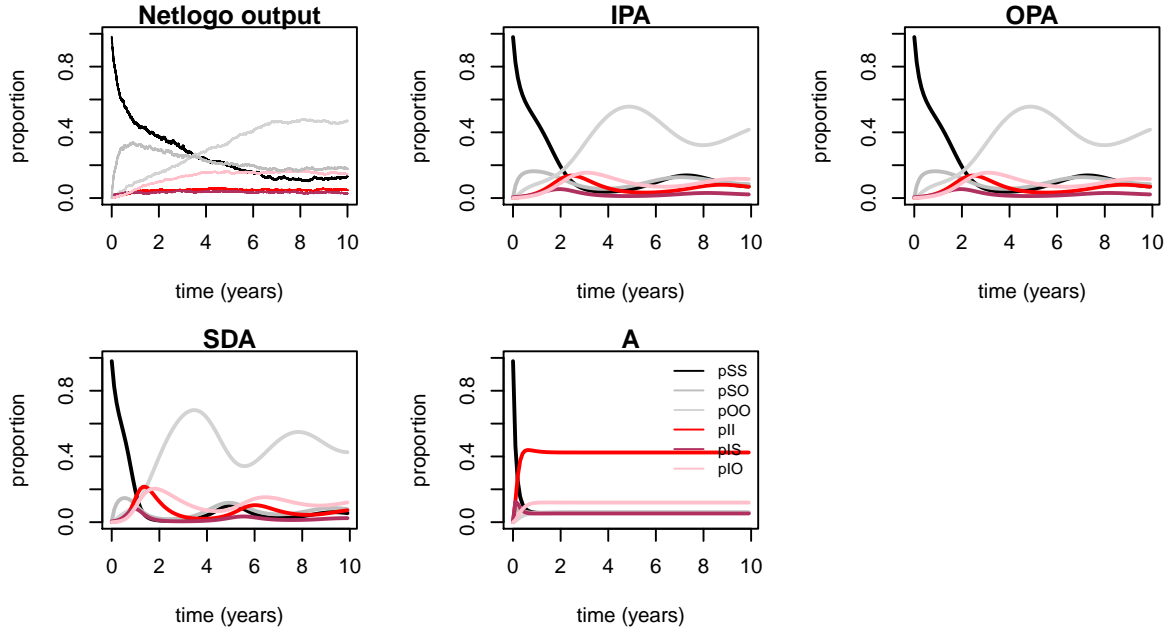


Figure 4: Generated by `Local_SI_Analysis.R`. This figure compares the output of `Local SI.nlogo` to the numerical solutions of the pair equations 7-12 using four different closures. A different closure may be needed to produce better agreement, or a system of triplet equations might be derived for better accuracy.

4 Additional reading

We will update this section with recommendations of required reading in a future version of this document.

References

- Boots, M. and A. Sasaki. 1999. ‘Small worlds’ and the evolution of virulence: infection occurs locally and at a distance. *Proc R Soc Lond B* 266: 1933-1938.
- Gillespie, D. T. 1977. Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry* 81(25): 2340-2361.
- Matsuda, H., N. Ogita, A. Sasaki, and K. Sato. 1992. Statistical Mechanics of Population: The lattice lotka-volterra model. *Progress of Theoretical Physical* 88(6): 1035-1049.
- Müller, J. and C. Kuttler, 1995. *Methods and Models in Mathematical Biology: Deterministic and Stochastic Approaches*. Lecture Notes on Mathematical Modelling in the Life Sciences. Springer. Berlin.
- Sato, K., H. Matsuda, A. Sasaki. 1994. Pathogen invasion and host extinction in lattice structured populations. *Journal of Mathematical Biology* 32: 251-268.
- Wikipedia, 2019. Paradox of enrichment. https://en.wikipedia.org/wiki/Paradox_of_enrichment.

A.1 Appendix

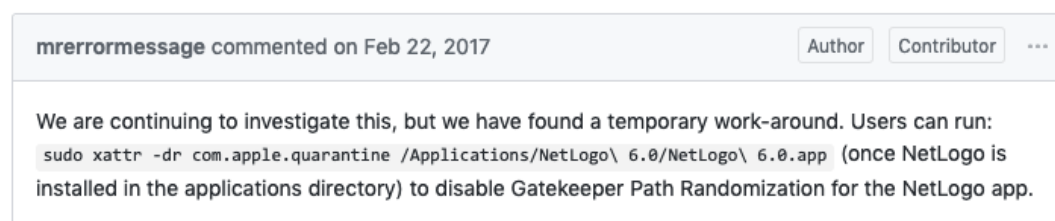
A.1.1 Software requirements

To complete all aspects of this workshop you will need the following software:

- **Netlogo.** See here <https://ccl.northwestern.edu/netlogo/download.shtml> for download information. nlogo files were written in Version 6.0.4.
- **R.** See here <https://www.r-project.org/> for download information. Files are compatible with version 3.5.1 “*Feather Spray*”. Some R files were written using RStudio as an interface, but RStudio is optional, not mandatory. The following packages need to be installed: **deSolve** and **GillespieSSA**.

A.1.2 Background on software and package choices and installation problems

Using a Macbook Air (2015) operating on Mojave macOS 10.14.6, I began working on these workshop materials by searching for an R package to simulate an ABM. I package I found was **RNetLogo**. This package is dependent on a **NetLogo**, so I updated to the most recent NetLogo (6.0.4). I had problems with this install, but eventually I found this solution Grider (2017). Specifically, I opened a terminal window and typed the command as described here:



RNetLogo is also dependent on the package **rjava**. I continued to have problems related to my specific operating system and the **rjava** package, in a workshop setting with many different types of laptops it would not be possible to troubleshoot problems, so I had to abandoned **RNetLogo**. Later, on twitter the recommendation was made to use the package **nlrx**, but at this point, the Netlogo to R connection is only a minor component of the necessary work and I decided to work with each software separately rather spend more time on automating the crosstalk between R and NetLogo.

A limitation of **NetLogo** is that every event is scheduled per iteration/time step, however, the Gillespie algorithm requires that the next event occur after variable amounts of time and this issue is discussed in Warnke et al. 2016. For **Netlogo** the recommendation was to install and work with the **time** extension (Sheppard and Railsback, 2015). Details of how to install are available from Sheppard and Railsback (2015) (specifically, download, name ‘time?’ and drop into extensions folder). The examples related to the **time** extension for **Netlogo** do not work for my installation of **NetLogo 6.0.4**. The Netlogo examples from Warnke et al. (2016) did not work either due to the same issue. Next, I read that **BehaviorSpace** (necessary to output the results of the ABM from **NetLogo**) is not compatible with the **time** extension.

Following these issues, I decided to track time as a variable and export it with **BehaviorSpace** bypassing he need to use the **time** extension. The downside of this approach is that time is nonlinear. Specifically, the visualizations in **NetLogo** show time progressing linearly, but in actuality, the amount of time that has passed between events is not constant. I am happy with this approach as the only dependency is on **Netlogo**.

A.1.3 Debugging in NetLogo

I have yet to really figure out how to debug in NetLogo. My current approach is to write in lines of code to `show X` where `X` is a parameter or `ask turtle 1 show X` if `X` is a turtles-own property. These values will then print into the Command Center window of the NetLogo interface.

References

- Grider, R (mrerrormessage). 2017. Ensure NetLogo will open in Mac OS X Sierra #1302. <https://github.com/NetLogo/NetLogo/issues/1302>
- Sheppard, C. J. R., S. Railsback, S. 2015. Time Extension for NetLogo (version 1.2). <https://github.com/colinsheppard/time>
- Warnke, T., O. Rienhardt, A. Uhrmacher. 2016. Population-based CTMCS and agent-based models. Proceedings of the 2016 Winter Simulation Conference. <https://ccl.northwestern.edu/2017/6.pdf>