Demographic, Stochastic SIR

Drew

July 7, 2014

1 Introduction

This document is about a little SIR simulation along the lines of the time-dependent infection rate example from Wearing and others[3]. We'll use the Semi-Markov library[2] to do the simulation.

2 ODE Model

The model, as given in Wearing, is an SIR with birth and death. Using $N = S + I + R$, μ as a death rate, γ as a recovery rate, and $\beta I/N$ as a freqency-dependent infection rate,

$$\frac{dS}{dt} = B - \frac{\beta I}{N} S - \mu S \tag{1}$$

$$\frac{dI}{dt} = \frac{\beta I}{N} S - \gamma I - \mu I \tag{2}$$

$$\frac{dR}{dt} = \gamma I - \mu R. \tag{3}$$

The crude birth rate, B, is an inhomgeneous term which fixes asymptotic population size by breaking scaling invariance, which you can see by adding the equations,

$$\frac{dN}{dt} = B - \mu N. \tag{4}$$

At long times and for fixed β,

$$N = B/\mu \tag{5}$$

$$S = \frac{(\mu+\gamma)B}{\beta\mu} \tag{6}$$

$$I = \frac{(\beta-\mu-\gamma)B}{\beta(\mu+\gamma)} \tag{7}$$

$$R = \gamma I/\mu \tag{8}$$

The model seems well-behaved. We are interested in adding a seasonal dependency to this model, so $\beta = \beta_0(1 + \beta_1 \cos(2\pi(t - t_i)))$. Time is measured in years, here. The values used in the example are

B	1/70 per year
β	400 per year
μ	1/70 per year
γ	365/14 per year.

Note that $N = 1$ with the choice of B/μ. Given these values, the Eqs. 5–8 predict $S = 0.07$, $I = 0.0005$, $R = .93$, which looks like what is in the graph called "$R(0) = 15.33$."

3 Stochastic Model

3.1 Model Definition

For background on stochastic SIR, see Allen's chapter[1]. Every transition in the stochastic model is exponential, which makes for a simple system. There are three places, S, I, and R, with a transition for each term of the differential equation. The GSPN is shown in Fig. 1. Working from the figure, we list our transitions.

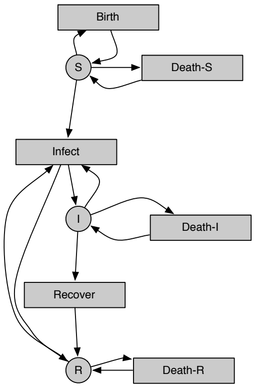

Figure 1: The GSPN for the stochastic model. Note that the infection transition includes inputs from all of S, I, and R because it depends on the total N individuals.

Transition	Hazard Rate
birth	B
death-s	μS
death-i	μI
death-r	μR
infect	$\beta SI/(S+I+R)$
recover	γI

3.2 Distribution for Seasonal Hazard Rate

The only complication is that the hazard rate will be seasonal,

$$\beta(t) = \beta_0(1 + \beta_1 \cos(2\pi(t - t_i))), \tag{9}$$

where t_i is an offset into the year and $\beta_1 < 1$. It is likely there will be enough events per year that we can safely approximate this piecewise, but let's start with the exact distribution because we can. We can write this as

$$\beta(t) = \beta_0(1 + \beta_1[\cos(2\pi t)\cos(2\pi t_i) + \sin(2\pi t)\sin(2\pi t_i)]). \tag{10}$$

This hazard depends on the absolute system time but does not depend on when an individual entered the susceptible state. We construct the distribution for this transition, from the hazard rate, as

$$F(t, t_0) = 1 - e^{-\int_{t_0}^{t} \beta_0(1+\beta_1\cos(2\pi(s-t_i)))ds}, \tag{11}$$

which we will need to invert, $U = F(t, t_0)$, in order to sample. Looking just at the integral,

$$\int_{t_0}^{t} \beta_0(1 + \beta_1 \cos(2\pi(s - t_i))) = \beta_0 \left((t - t_0) + \frac{\beta_1}{2\pi}\sin(2\pi(t - t_i)) - \frac{\beta_1}{2\pi}\sin(2\pi(t_0 - t_i)) \right) \tag{12}$$

If we define $t_0' = t_0 + \frac{\beta_1}{2\pi}\sin(2\pi(t_0 - t_i))$, then we have to solve

$$-\ln(1 - U)/\beta_0 + t_0' = t + \frac{\beta_1}{2\pi}\sin(2\pi(t - t_i)). \tag{13}$$

4 Sample Runs and Timing

Using the initial values from the Eq. 8, the graph in Fig. 2 shows a long time equilibrium value which is the same as that for the ODE.

Runtime for the model, covering 30 years, is shown in the table below. The results come from `python runner.py -time -runcnt 10`.

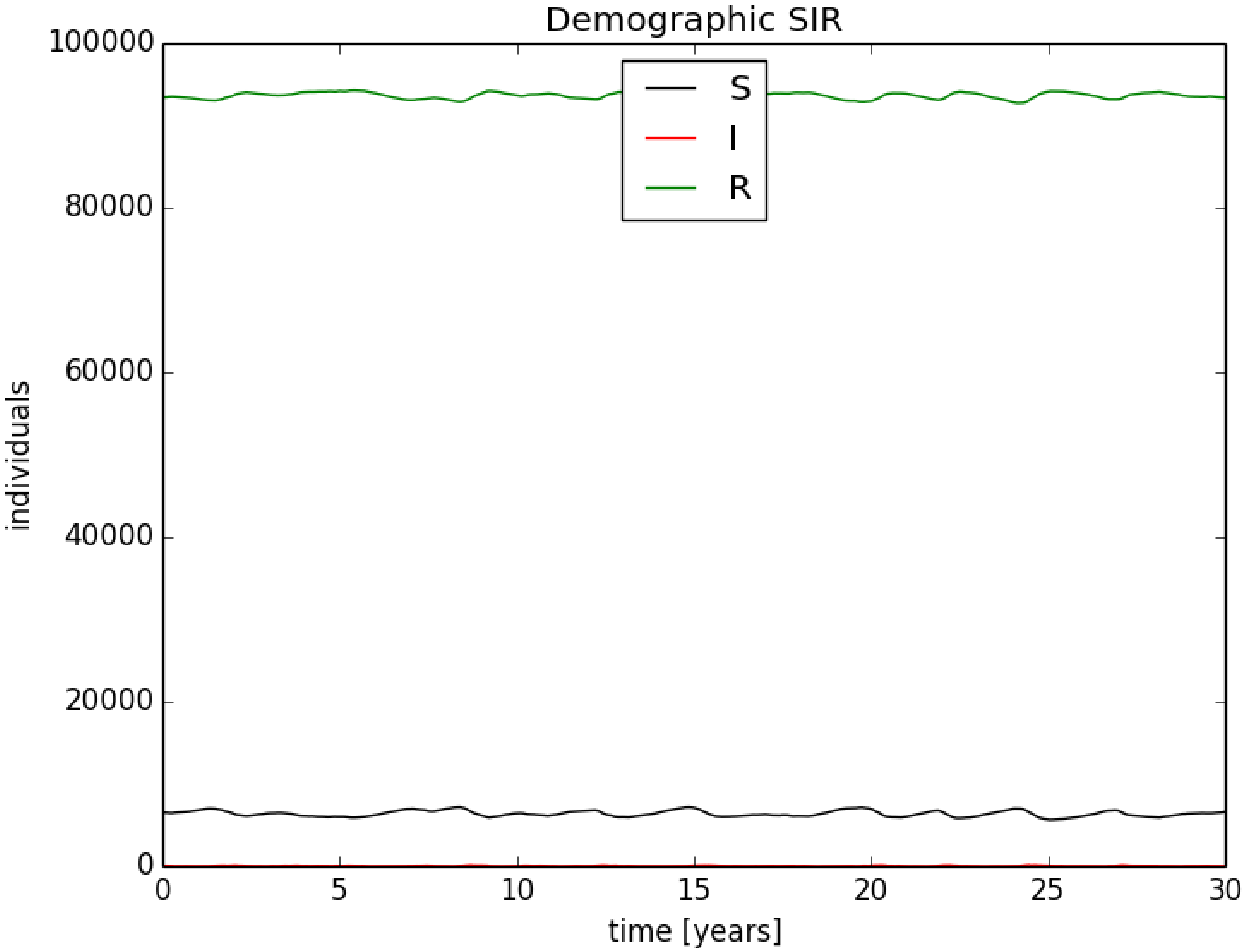

Figure 2: Initial values from ODE are the same as the long-time equilibrium of this stochastic model of 100,000 individuals. Here there is no seasonality.

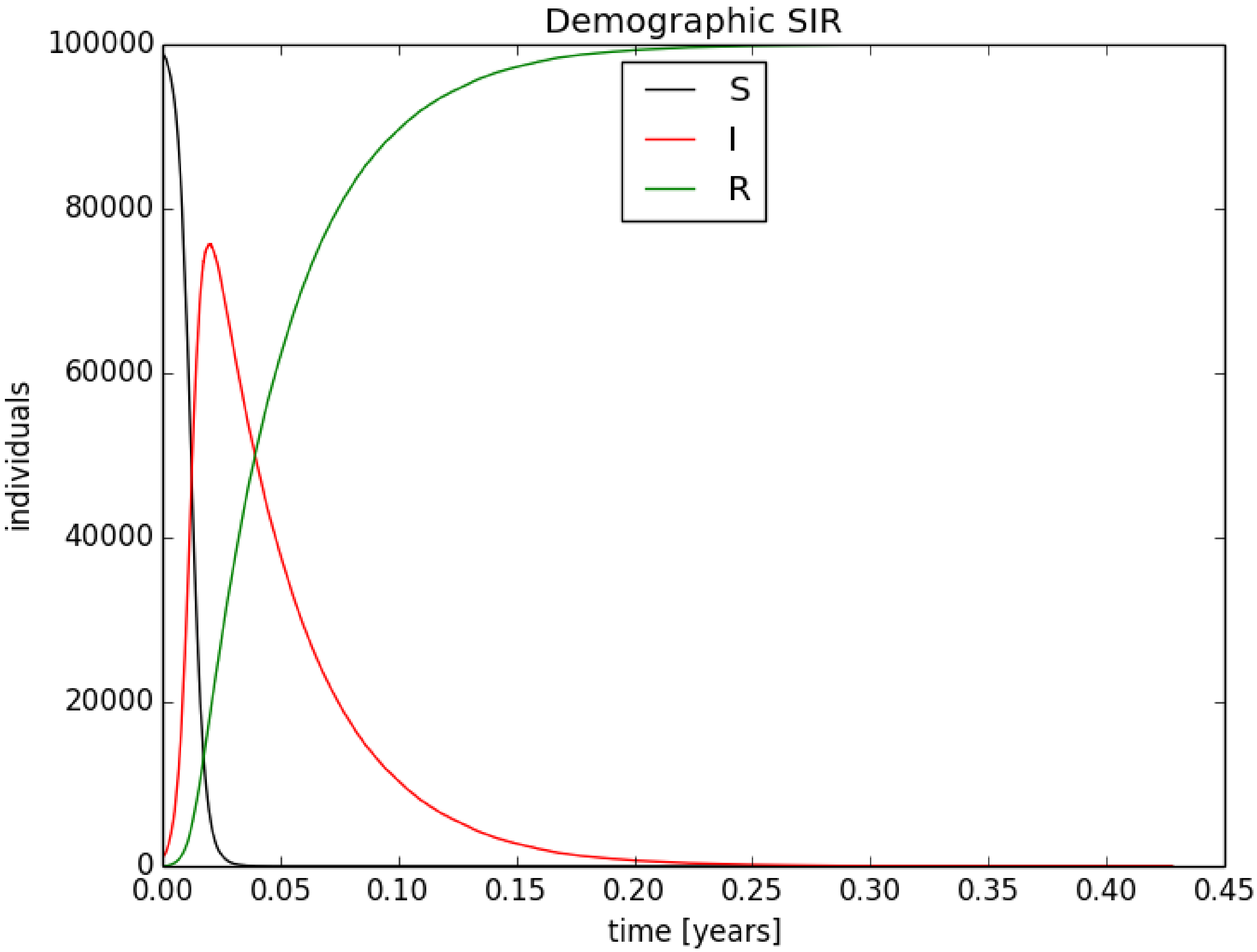

Figure 3: Another trajectory, this time for no birth or death. It looks as we hope for SIR.

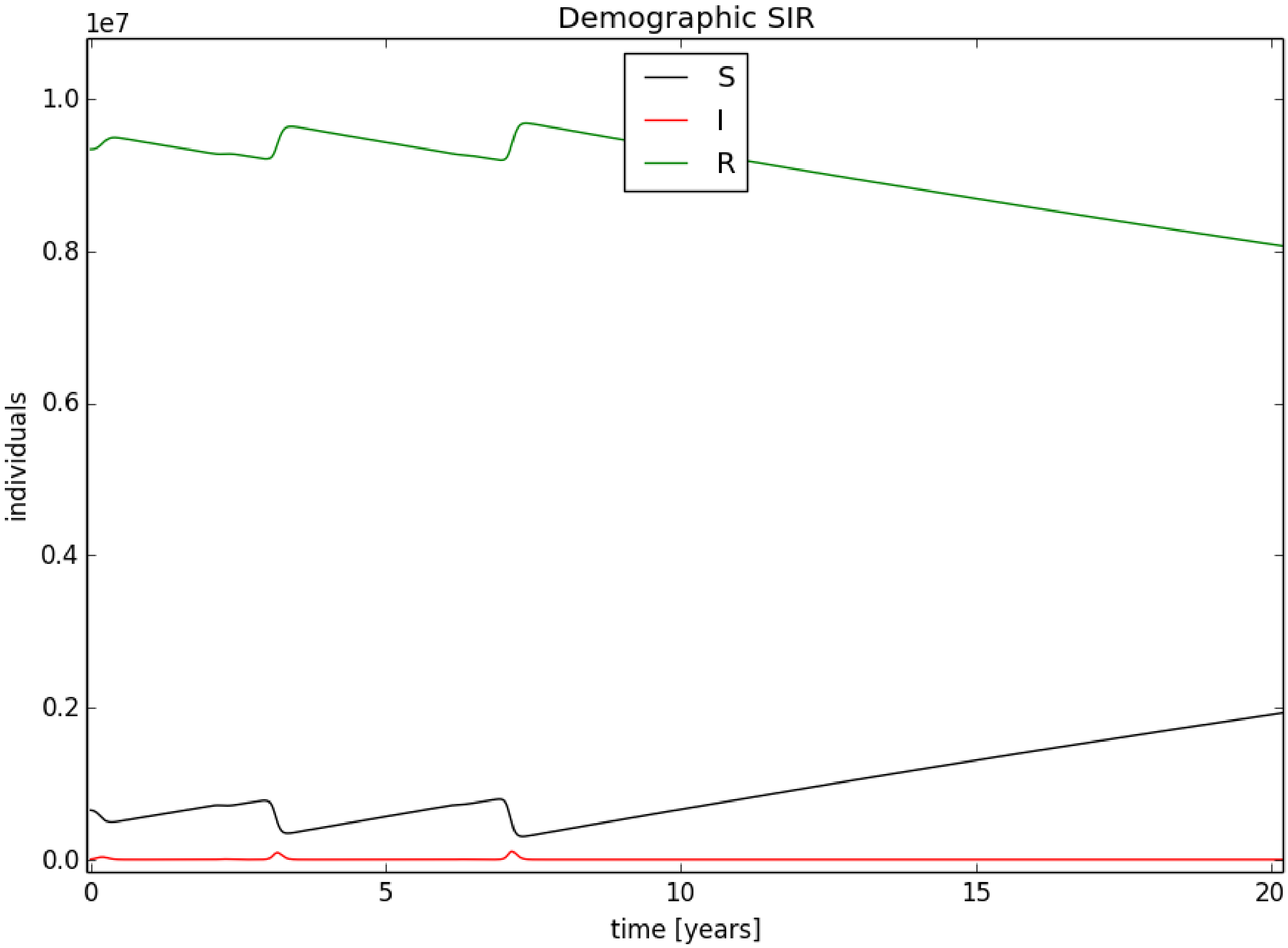

Figure 4: When we go to 10,000,000 individuals, the results look like the ODE.

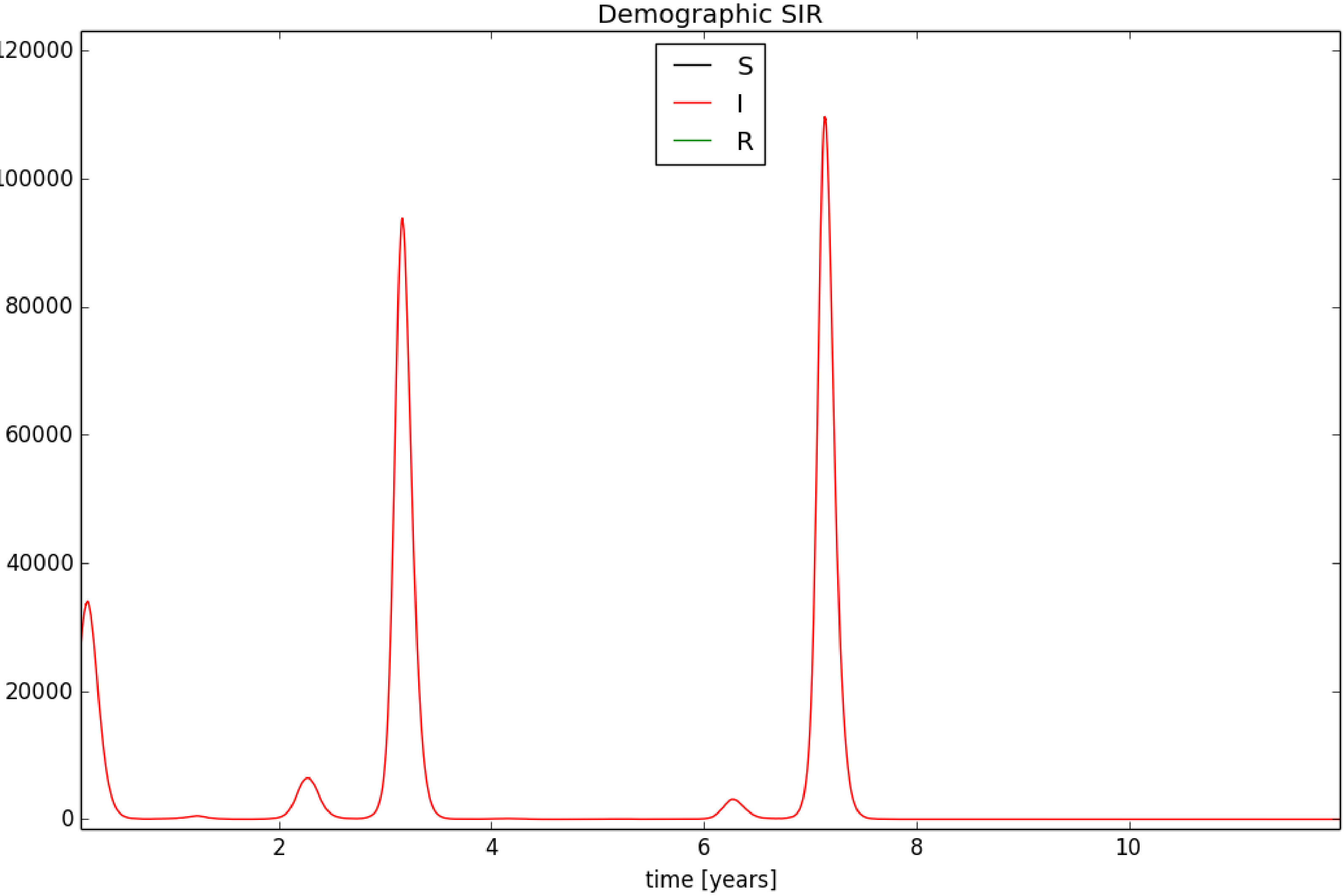

Figure 5: A closeup of the smoothed trajectory for 10,000,000 individuals shows how seasonality affects the number of infected.

# individuals	wall clock seconds
5000	0.013
10,000	0.025
20,000	0.050
50,000	0.14
100,000	0.37
200,000	1.0
500,000	5.3
1,000,000	19

These values assume a single thread, but the code is multi-threaded using the `-j` flag. For instance, running 100, 30-year trajectories of 200,000 individuals on my 4-core machine takes about 30 seconds.

5 Running the Executable

Command-line options include all of the parameters to the system, plus the following:

`-j` The number of threads to use.

`--runcnt` How many times to run the simulation.

`-s` The total number of individuals. Not the number of susceptibles, surprisingly. Want that changed?

`-i` Total number of infecteds.

`-r` Total number of recovereds.

`--seed` Starting random seed. Multithreaded runs use a seed per thread, so don't just increment a single seed.

`--endtime` How many years to run.

`--exacttraj` There are two observers of the trajectory. One saves every step. The inexact one only saves steps when they change .1% of the value.

`--exactinfect` Whether to use the time-dependent infection rate or the inexact version, which uses a piecewise approximation. This exists so that we can put an upper bound on the error to the obviously good choice of a piecewise infection rate.

`--datafile` Filename to which to save. This is overwritten each time. We can change that, too.

`--loglevel` Whether to print logging information. You can quiet the program significantly with `--loglevel=error`.

`--info` Show how the program was built, meaning its git repository and makefile.

Files in the directory:

default_parser.py Puts some default arguments into Python's argparse module.

freqsir.* This document.

getgit.py Retrieves the git repository information from the current directory and places it into a C++ header file so the program can know where it came from.

hdf_file.{hpp,cpp} The file format uses HDF5. This class uses a C++ PIMPL idiom. Note the mutex which guarantees only one writer at a time, so multiple threads can hold this object.

main.cpp The C++ main().

Makefile The SMVHIDELOG option removes all tracing code, which is about ten percent or more of the run time. See this file for the list of required libraries.

quick.py Plotting trajectories using Matplotlib.

runner.py Runs timing. Also a demonstration of how to read the HDF5 file.

seasonal.{cpp,hpp} Implements a statistical distribution whose hazard rate is the seasonal hazard for infection. Uses GSL's root-finding methods to solve for the sample.

sir_exp.{cpp,hpp} This is the GSPN model! Everything scientific is in this file, between lines 100 and 400.

References

[1] Linda J. S. Allen. An Introduction to Stochastic Epidemic Models. In Fred Brauer, Pauline van den Driessche, and Jianhong Wu, editors, *Mathematical Epidemiology*, volume 1945 of *Lecture Notes in Mathematics*, chapter 3, pages 81–130. Springer Berlin / Heidelberg, 2008.

[2] Andrew J. Dolgert and David J. Schneider. Semi-markov library, 2014.

[3] Helen J. Wearing, John M. Drake, and Aaron A. King. Simulating Epidemics in R. Technical report, 2012.