

Stochastic VCOM

Sean L. Wu^{1*}

¹Divisions of Biostatistics & Epidemiology, University of California, Berkeley

*Correspondence: slwu89@berkeley.edu, slwood89@gmail.com

January 2020

1 Introduction

To translate the **VCOM** ODE model into a stochastic model in a mathematically sound and rigorous manner, we first reformulated the system of ODEs into a set of difference equations with the same qualitative dynamics. To do so, we imagined imagine that the ODEs describe a continuous-time Markov chain (CTMC); that is, the right hand side of the ODE system describes various processes, each of which compete to induce a change in state-space. As a refresher, we list the original **VCOM** ODEs as follows:

$$\begin{aligned}\frac{dE}{dt} &= \beta V - \mu_E \left(1 + \frac{E+L}{K}\right) E - \frac{E}{d_E} \\ \frac{dL}{dt} &= \frac{E}{d_E} - \mu_L \left(1 + \gamma \frac{E+L}{K}\right) L - \frac{L}{d_L} \\ \frac{dP}{dt} &= \frac{L}{d_L} - \mu_P P - \frac{P}{d_P} \\ \frac{dS_V}{dt} &= \frac{1}{2} \frac{P}{d_P} - \lambda_V S_V - \mu_V S_V \\ \frac{dE_V}{dt} &= \lambda_V S_V - \frac{E_V}{EIP} - \mu_V E_V \\ \frac{dI_V}{dt} &= \frac{E_V}{EIP} - \mu_V I_V\end{aligned}\tag{1}$$

Where the total adult female population is $V = S_V + E_V + I_V$, and λ_V represents the force of infection (FOI) on mosquitoes. The equations are identical except for that the EIP is now exponentially distributed because the derivation of the stochastic model is more straightforward this way. Delay differential equation models (DDEs) will lead to Markov chains with lag and are cumbersome to deal with. If a more accurate representation of EIP is desired, it will be easiest to use an Erlang-distributed delay.

2 ODEs as a CTMC

Now to begin developing our stochastic model, let us imagine this set of ODEs as describing an approximate mean behavior of a CTMC (it will not be exact due to non-linearities in the larval death rates). Now, each individual term in Equation 1 can be interpreted as the rate of of Poisson Process that tells us how fast each event occurs. The events and associated rates are listed in Table 1:

Event	Rate
Oviposition	βV
Death of Early-stage Larval Instars	$\mu_E \left(1 + \frac{E+L}{K}\right) E$
Advancement of Early-stage Larval Instars	$\frac{E}{d_E}$
Death of Late-stage Larval Instars	$\mu_L \left(1 + \gamma \frac{E+L}{K}\right) L$
Advancement of Late-stage Larval Instars	$\frac{L}{d_L}$
Death of Pupae	$\mu_P P$
Emergence of Pupae	$\frac{P}{d_P}$
Emergence of Adult Females	$\frac{1}{2} \frac{P}{d_P}$
Infection of Susceptible Adult Females	$\lambda_V S_V$
Death of Susceptible Adult Females	$\mu_V S_V$
Completion of EIP	$\frac{E_V}{EIP}$
Death of Incubating Adult Females	$\mu_V E_V$
Death of Infectious Adult Females	$\mu_V I_V$

Table 1: Events & rates

2.1 Deterministic Euler Approximation

One way to simulate from a CTMC is to do an Euler approximation. In order to do this, we set some finite time step Δt . We start by doing a deterministic Euler approximation that gives the expected change in each state over the time step.

$$\begin{aligned}
\Delta E &= (\beta V \Delta t) - \left(1 - e^{-(\mu_E(1 + \frac{E+L}{K}) + 1/d_E)\Delta t}\right) E \\
\Delta L &= \left(1 - e^{-(\mu_E(1 + \frac{E+L}{K}) + 1/d_E)\Delta t}\right) \left(\frac{1/d_E}{\mu_E(1 + \frac{E+L}{K}) + 1/d_E}\right) E - \left(1 - e^{-(\mu_L(1 + \gamma \frac{E+L}{K}) + 1/d_L)\Delta t}\right) L \\
\Delta P &= \left(1 - e^{-(\mu_L(1 + \gamma \frac{E+L}{K}) + 1/d_L)\Delta t}\right) \left(\frac{1/d_L}{\mu_L(1 + \gamma \frac{E+L}{K}) + 1/d_L}\right) L - \left(1 - e^{-(\mu_P + 1/d_P)\Delta t}\right) P \\
\Delta S_V &= \left(1 - e^{-(\mu_P + 1/d_P)\Delta t}\right) \left(\frac{1/2d_P}{\mu_P + 1/d_P}\right) P - \left(1 - e^{-(\mu_V + \lambda_V)\Delta t}\right) S_V \\
\Delta E_V &= \left(1 - e^{-(\mu_V + \lambda_V)\Delta t}\right) \left(\frac{\lambda_V}{\mu_V + \lambda_V}\right) S_V - \left(1 - e^{-(\mu_V + 1/EIP)\Delta t}\right) E_V \\
\Delta I_V &= \left(1 - e^{-(\mu_V + 1/EIP)\Delta t}\right) \left(\frac{1/EIP}{\mu_V + 1/EIP}\right) E_V - \left(1 - e^{-\mu_V \Delta t}\right) I_V
\end{aligned} \tag{2}$$

The set of equations can be interpreted by noting that all of the $1 - e^{-h\Delta t}$ terms tell us the Euler approximation of how many individuals in that state experience *any* event over the time step. It is not exact because λ could be changing over Δt , but we hope that it's roughly linear for small-ish time steps. That quantity appears in the line below multiplied by a fraction to give the number of individuals that experienced the event of interest (here advance to the next state rather than dying). For example, $\left(1 - e^{-(\mu_V + \lambda_V)\Delta t}\right) S_V$ is the total number of susceptible adult females that are *either* infected or die over the time step. In the equation for ΔE_V , some fraction of those individuals $\left(\frac{\lambda_V}{\mu_V + \lambda_V}\right)$, namely, the ones who were infected, are added to the incubating mosquitoes.

2.1.1 Equilibrium Solution to Deterministic Euler Approximation

Solving the equilibrium for the discrete time Euler model is a greater challenge than for the ODE model, due to additional non-linearities introduced by the exponential functions. We will assume \bar{I}_V is known; for transmission modelling, this is useful because given assumptions about a few entomological parameters, \bar{I}_V can be solved for if we know the EIR.

$$\text{EIR} = a \frac{\bar{I}_V}{H} \quad (3)$$

If the human biting rate a and human population size H are known, \bar{I}_V can be easily solved for. After we have the value of \bar{I}_V we set the RHS of Equation 2 equal to zero. We can only derive analytic solutions for $\bar{P}, \bar{S}_V, \bar{E}_V$ as the remaining equations are transcendental.

$$\begin{aligned} \bar{E}_V &= \frac{\bar{I}_V (1 - e^{-\mu_V \Delta t})}{(1 - e^{-(\mu_V + 1/EIP)\Delta t}) \left(\frac{1/EIP}{\mu_V + 1/EIP} \right)} \\ \bar{S}_V &= \frac{\bar{I}_V (1 - e^{-\mu_V \Delta t}) + \frac{(1 - e^{-(\mu_V + 1/EIP)\Delta t}) \left(\frac{\mu_V}{\mu_V + 1/EIP} \right) \bar{I}_V (1 - e^{-\mu_V \Delta t})}{(1 - e^{-(\mu_V + 1/EIP)\Delta t}) \left(\frac{1/EIP}{\mu_V + 1/EIP} \right)}}{(1 - e^{-(\mu_V + \lambda_V)\Delta t}) \left(\frac{\lambda_V}{\mu_V + \lambda_V} \right)} \\ \bar{P} &= \frac{2\bar{S}_V (1 - e^{-(\mu_V + \lambda_V)\Delta t})}{(1 - e^{-(\mu_P + 1/d_P)\Delta t}) \left(\frac{1/d_P}{\mu_P + 1/d_P} \right)} \end{aligned} \quad (4)$$

To solve for \bar{E}, \bar{L}, K we derived approximate values from the equilibrium solutions of the ODE model. We then generate 250 random starting points for numerical optimization by Latin Hypercube sampling in a grid around these approximate values, where the bounds of the grid are given by $\pm 12.5\%$ of the absolute value of those parameters in each dimension. We then take each starting point as an initial value for linearly-constrained BFGS optimization (to respect positivity for all parameters as well as $L > E$) and pick the final parameter set that minimizes the sum of squared errors of $(\Delta E, \Delta L)$ away from 0 (at equilibrium the difference equations should be equal to 0). It should be noted that the discretization Δt affects the equilibrium solutions.

2.2 Stochastic Euler Approximation

When formulating the stochastic model as a CTMC we treat all rates (see Table 1) as instantaneous hazard functions, then assume the integrated hazard is a linear function over the chosen Δt (this is the Euler approximation). These approximate integrated hazards give us the means to construct a Multinomial probability distribution over events each mosquito may experience in a given compartment (including the event that no events occur to that individual). We also note that this stochastic Euler method precludes a mosquito from making multiple jumps in a single time step, whereas over the same period of time in the exact CTMC such a series of jumps is possible (albeit with very small probability). In fact from a certain point of view, the stochastic Euler method is not a discretization at all, but rather creating an entirely new discrete-time Markov chain model (DTMC) whose dynamics converge to the CTMC as $\Delta t \rightarrow 0$. By coupling Multinomial jump processes for each compartment with an extra Poisson process for oviposition we generate such a DTMC model.

To simulate approximate trajectories from the stochastic model we use a Euler-multinomial scheme to sample the system change over Δt . For a number of individuals N_i in compartment i at time t , we sample ΔN_{ij} , where $j \in (1, \dots, k)$ are all of the possible destinations from that compartment an individual could

jump to in the time interval $[t, t + \Delta t)$. Given a probability vector $\{p_{i \rightarrow j}\}$ describing the probability for individuals in i to jump to each destination j , the random vector describing where individuals jump to is multinomially distributed as (note the first element describes those individuals who stay in compartment i):

$$(N_i - \sum_j \Delta N_{ij}, \Delta N_{i1}, \dots, \Delta N_{ik}) \sim \text{Multinomial}(N_i; p_{i \rightarrow i}, p_{i \rightarrow 1}, \dots, p_{i \rightarrow k}) \quad (5)$$

The probability vector can be calculated by linearly approximating the integrated hazards, as noted earlier. This gives the following equations for the probability of staying, and the probability to go to a specific destination compartment, conditional on leaving.

$$\begin{aligned} p_{i \rightarrow i} &= e^{-\sum_j \mu_{ij} \Delta t} \\ p_{i \rightarrow j} &= \left(1 - e^{-\sum_j \mu_{ij} \Delta t}\right) \left(\frac{\mu_{ij}}{\sum_j \mu_{ij}}\right) \end{aligned} \quad (6)$$

For example, for mosquitoes in compartment S_V the possible transition probabilities are $p_{S_V \rightarrow E_V}$, if they become infected, and $p_{S_V \rightarrow D}$, if they die. Of course $p_{S_V \rightarrow S_V}$ gives the probability that they remain (nothing happens).

$$\begin{aligned} p_{S_V \rightarrow S_V} &= e^{-(\mu_V + \lambda_V) \Delta t} \\ p_{S_V \rightarrow E_V} &= \left(1 - e^{-(\mu_V + \lambda_V) \Delta t}\right) \left(\frac{\lambda_V}{\mu_V + \lambda_V}\right) \\ p_{S_V \rightarrow D} &= \left(1 - e^{-(\mu_V + \lambda_V) \Delta t}\right) \left(\frac{\mu_V}{\mu_V + \lambda_V}\right) \end{aligned} \quad (7)$$

Egg laying is treated as a special case, as we assume each mosquito lays eggs according to a Poisson process, such that over Δt the total number of eggs laid is Poisson with mean $(\beta N \Delta t)$.

3 Simulation Code

The repository **s-vcom** contains both **R** and **C++** code to run the deterministic and stochastic models with ITN/IRS. All of the code to run the simulation and calculate equilibrium values are in the folder **sim-src**. The file **mosquito-equilibrium.R** contains a function **calc_eq** which calculates the equilibrium values as described earlier. It requires as input a vector of parameters **theta**, the size of the time step Δt , the size of the infectious mosquito population $\overline{I_V}$, and the force of infection on mosquitoes λ_V . It returns equilibrium values for $\overline{E_V}, \overline{S_V}, \overline{P}, \overline{L}, \overline{E}, K$.

To run the simulation in **R**, code is provided in both **mosquito-stochastic.R** and **mosquito-deterministic.R**. Both files provide the same two functions, but for the stochastic and deterministic simulations, respectively. The function **make_node** generates a **R** list type of data object that contains all of the state necessary for the simulation. The function **euler_step** takes as input the list and updates the dynamics over a single time step. Examples of how to use both are given in the script **example.R** in the root directory of the repository.

The **C++** file **mosquito-both.cpp** contains deterministic and stochastic simulation code. The code defines a template class, **mosquito_habitat**, with full specialization for types **double** and **int**; these correspond to simulating the deterministic and stochastic system, respectively. It exposes two functions to **R**, **cpp_deterministic** and **cpp_stochastic**, which return a matrix where each column is a life stage (E, L, P, S_V, E_V, I_V) and rows are time steps. The **C++** code relies on the built-in **R** random number generator so that runs can be reproducible via calls to **set.seed**. The speed of the **C++** is useful for

Monte Carlo simulation, parameter fitting, and sensitivity analysis. An example of output produced by the stochastic simulation is shown in Figure 1.

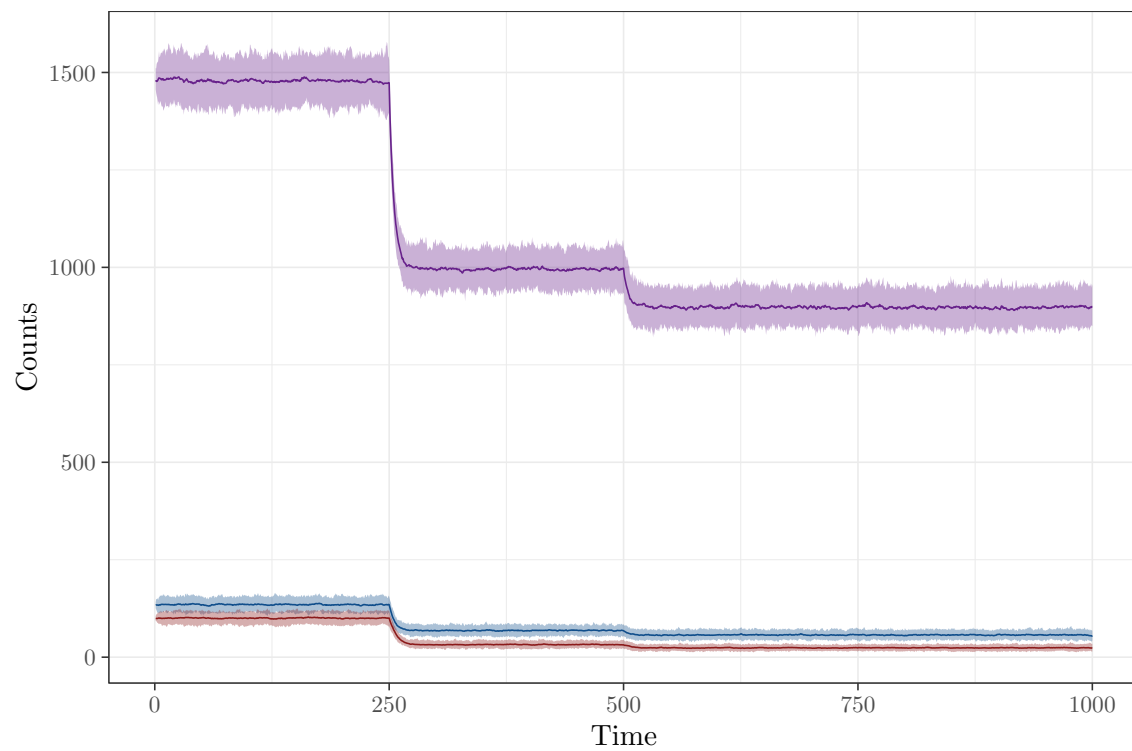


Figure 1: Output of 100 stochastic **s-vcom** simulations, means are dark lines and 95% quantiles are in shaded area