disaggregated simulation algorithm

Recall that the "big picture" algorithm look like this:

```
tmax <- t0 + dt
while (tnow < tmax) {

    simulate_mosquitio_pop(mosquito_pop, tnow, dt)
    simulate_human_pop(human_pop, tnow, dt)

    bloodmeal(human_pop, mosquito_pop, tnow, dt)

    tnow += dt
}
```

mosquito simulation on first time step

Consider the problem of starting with simulation of the mosquito population `simulate_mosquitio_pop`, at the initial time point. Then there is no input of bites yet, so the algorithm can directly begin to simulation S_V, E_V, I_V. Let biting input be stored in a queue data structure called `H2M_bloodmeal` (human to mosquito transmission).

Because there is no input of bites which would cause an $S \to E$ event, we can directly simulate the events in the system.

If E_V is nonzero, we queue $E \to I$ events for each exposed mosquito, call the queue data structure which stores these events `E2I_mosquito`. This is done "before simulation starts". The time that each of these mosquitos will become infected is a realization of the random variable $Unif(-\tau_{EIP}, 0) + \tau_{EIP}$. A draw is taken for each, and it is added to `E2I_mosquito`. Values are drawn from this random variable because if a mosquito was incubating at time 0, it could not have become infected at a time less than $-\tau_{EIP}$, otherwise it would have been infectious at time 0. Therefore, under the assumption that the model is (except for the deterministic delays), a set of coupled Poisson processes (because the infection time would have arisen from one), the actual infection time is uniformly distributed along the interval $(-\tau_{EIP}, 0]$. Therefore the $E \to I$ transition will be that initial infection time plus τ_{EIP}.

After this sampling of initial conditions, the mosquito population can be simulated along the time interval from $[t_0, t_0 + \Delta t)$. Henceforth let $t_0 + \Delta t = t_1$, so that generally the i^{th} iteration is denoted by its right open endpoint t_i. Deaths from each compartment occur at per-capita intensity g, so for example the Poisson process accounting for deaths from S_V has intensity gS_v. Emergence is even simpler, as we assume it occurs with state-independent intensity λ. If the next event is an $E \to I$ transition, we must update slightly more carefully, by removing it from the head of the queue `E2I_mosquito` during the state update. If the event is a death from E_V, then a random element from `E2I_mosquito` is removed (note that the size of the queue is always equal to the value of E_V). The whole system can be simulated by Anderson's "modified next reaction method" (MNRM), a technique to simulate Markovian or non-Markovian continuous time discrete event stochastic systems. The mosquito state is simulated using the MNRM until the next event time would exceed t_1, the end of the TWICE interval.

During this whole time we must also accumulate the piecewise constant (cadlag) trajectory describing the changes in $\tilde{S}_V$ over this initial time interval $[t_0, t_1)$. Note also that mosquitos in this trajectory represent mosquitos which may be either in S_V or E_V (hence the tilde notation), because the bloodmeal events have not been sampled yet, though we are always guaranteed that at t_0 all the mosquitos are in $\tilde{S}_V$. Also note that emergence events which increase $\tilde{S}_V$ by 1 will increase the population of susceptible mosquitos, although deaths, which decrease $\tilde{S}_V$ by 1 may either occur to a susceptible mosquito or an infected mosquito. This will be relevant when sampling bloodmeal events later.

The mosquitos must also output the trajectory of I_V over this interval, needed for the computation of bloodmeals causing infection in humans.

human simulation on first time step

Because we do not model deaths or births in the human component, a description of the algorithm for `simulate_human_pop` is even simpler. Again, for non-zero E_H, we do a similar process as to the initial mosquito incubating individuals to sample the initial state. The difference is that the random variable which is sampled and added to `E2I_human` is now described by $Unif(-\tau_{LEP}, 0) + \tau_{LEP}$, following the same argument. On the first time step there are no queued infection events known to occur over this step, so the queue data structure `M2H_bloodmeal` is empty.

The only Poisson process in the human model is that governing the $I \to S$ transition, firing at intensity rI_H. It competes with the deterministic events queued in `E2I_human`. The same update procedure based on the MNRM used for the mosquitos can be used to simulate the humans.

Like the mosquitos, the humans must output the trajectory of $\tilde{S}_H$ over this interval. They must also output the trajectory of X (prevalence), needed for the computation of bloodmeals causing infection in mosquitos.

bloodmeal algorithm on the first time step

infection in mosquitos $S_V \to E_V$ on first time step

We now need to introduce some notation. In the following notation, the subscript i will denote the right hand open endpoint for a piecewise constant interval in the input trajectory of susceptible mosquitos. Let $q_0 = t_0$, and q_1 is the right endpoint of the first interval. Let q_n denote the right hand open endpoint of the final interval, which will equal t_1. Now, let s_i denote the value of the state trajectory $\tilde{S}_V$ in this interval. Let $\tilde{r}_i$ denote the *putative risk set* over this interval, that is the number of susceptible mosquitos at the left hand closed endpoint q_{i-1}. Let r_i denote the *risk set* at the end of this interval, that is, the number of susceptible mosquitos at right hand open endpoint q_i.

On the first step $i = 1$ the putative risk set is $\tilde{r}_1 = s_1 = \tilde{S}_V(t), t \in [q_0, q_1)$. From the human module we know the value of prevalence X over that interval as well. Each mosquito in the putative risk set has probability p_i of taking a bloodmeal that results in infection during that interval, calaculated as:

$$p_i = 1 - \exp\left(-\int_{q_0}^{q_1} acX(s)ds\right)$$

Then the number of infections in mosquitos that become infected in the first interval is a draw from the random variable:

$$n_i \sim Binomial(\tilde{r}_i, p_i)$$

Then the risk set (number of mosquitos still susceptible) at the end of the interval (q_1) is $r_i = \tilde{r}_i - n_i$. Due to the assumption of the model as a set of competing Poisson processes, each of the n_i infection events occurs at a time uniformly distributed on the interval, that is $b_i \sim Unif(q_0, q_1$. Each of these bites must be added to the `H2M_bloodmeal` queue.

infection in mosquitos $S_V \to E_V$ on subsequent time steps

When $i > 1$, the algorithm is similar, but must account for the changing size of the risk set due to demographic events in the mosquito population, which are the jumps in $\tilde{S}_V$. At the moment in time q_{i-1} the trajectory

jumps from s_{i-1} to s_i, the difference is given by $s_i - s_{i-1}$. Because the model is a set of simple (not compound) counting processes, the difference can only take values of $+1$ or -1.

Let's see how this influences the size of the risk set. The size of the risk set *right before* the jump is r_{i-1}. If the jump took a value of $+1$ at q_{i-1}, then we know the risk set grew by 1, because we assume all mosquitos emerge susceptible, and $\tilde{r}_i = r_{i-1} + 1$. If the jump took a value of -1, then we have to sample to determine if the death occurred among the set of infected or susceptible mosquitoes at q_{i-1}. The probability the death occurred to a susceptible mosquito (i.e. a mosquito in the risk set) is r_{i-1}/s_{i-1}, and the probability it occurred to an infected mosquito is $1 - r_{i-1}/s_{i-1}$. Sample a Bernoulli random variate with those probabilities, if the death occurred to a susceptible mosquito, the risk set decreases by 1, that is $\tilde{r}_i = r_{i-1} - 1$. If the death occurred to an infected mosquito, the risk set does not decrease, that is $\tilde{r}_i = r_{i-1}$. Note that we must now "update the history", however we are tracking the overall trajectory of the model, and assign that death concretely to the susceptible or infected population as appropriate (i.e. S_V or E_V).

Now that we know the putative risk set, we can again calculate the per-capita probability of infection p_i, and sample n_i.

pseudocode

We show the algorithm below in pseudocode. Let `s` equal $\tilde{S}_V$, which can be linearly indexed by interval.

```
s_prev <- s[1]
r_prev <- 0
i <- 1
while (i <= n)

  s <- s[i]

  if s - s_prev == -1
    if rand() < r_prev/s_prev # death occured among susceptible
      rbar <- r_prev - 1
      update_history(...) # reassign this death event to decrease the S_{V} population
    else
      rbar <- r_prev
      update_history(...) # reassign this death event to decrease the E_{V} population
    end
  else if s - s_prev == 1
    rbar <- r_prev + 1
  else if s - s_prev == 0
    if i != 1
      error("all interior intervals must correspond to an emergence or death event")
    else
      rbar <- s
    end
  else
    error("jumps larger than one are not allowed")
  end

  p <- 1 - exp(-integrate(a*c*X(s),q[i-1],q[i]))
  n <- rbinomial(rbar, p)
  r <- rbar - n

  # sample actual infection times
  for i in 1:n
```

```
    btime <- runiform(q[i-1],q[i])
    push!(H2M_bloodmeal, btime)
  end

  r_prev <- r
  s_prev <- s
  i <- i + 1
end
```

infection in humans $S_H \to E_H$

The algorithm to compute the bloodmeal events in humans is considerably simpler, because in the simple model here demography in the human population is not modeled. Let us reuse the notation from the mosquito bloodmeal, except that s now refers to the state trajectory of $\tilde{S}_H$.

On the first interval ($i = 1$) the risk set in humans is $\tilde{r}_i = s_i$. The per-capita probability of infection is equal to:

$$p_i = 1 - \exp\left(-\int_{q_0}^{q_1} abI_V(s)/N ds\right)$$

Then the number of infections in humans over this interval is again:

$$n_i \sim Binomial(\tilde{r}_i, p_i)$$

And the risk set at the end of this interval is $r_i = \tilde{r}_i - n_i$. Again, each biting time is distributed according to $Unif(q_0, q_1$, and each bite time is added to the `M2H_bloodmeal` queue.

On subsequent time steps, we know that the trajectory $\tilde{S}_H$ changes only when the risk set expands due to recoveries from I_H. Therefore we update $\tilde{r}_i = r_{i-1} + 1$ at each jump time. We can again go through the above procedure to simulate bites causing infection over that interval until entire interval has been simulated. We outline the algorithm in the below pseudocode:

```
s_prev <- s[1]
r_prev <- 0
i <- 1
while (i <= n)

  s <- s[i]

  if s - s_prev == 0
    rbar <- s
  else if s - s_prev == 1
    rbar <- r_prev + 1
  else
    error("illegal jump detected")
  end

  p <- 1 - exp(-integrate(a*b*IV(s)/N,q[i-1],q[i]))
  n <- rbinomial(rbar, p)
  r <- rbar - n

  # sample actual infection times
```

```
  for i in 1:n
    btime <- runiform(q[i-1],q[i])
    push!(M2H_bloodmeal, btime)
  end

  r_prev <- r
  s_prev <- s
  i <- i + 1
end
```

mosquito simulation on subsequent time steps

Simulation of the mosquito population on TWICE steps after the first (i.e. on $[t_0+\Delta t, t_0+2\Delta t)$ and onward) is exactly as the first except that now the `H2M_bloodmeal` queue is not empty, so before the previous simulation begins we must update the mosquito state with those infection events that occurred during the previous TWICE step.

For each bite stored in the `H2M_bloodmeal` we must remove it, and queue an exposed to infectious transition in the `E2I_mosquito` queue at time $b_i + \tau_{EIP}$, and also increment the value of E_V at q_{i-1} by $+1$, and decrement S_V at q_{i-1} by -1. Note that at the end of this process the size of `E2I_mosquito` should equal E_V. After this simulation of demographic processes proceeds as before.

human simulation on subsequent time steps

Simulation of the human population is similar to the mosquito situation, except bites are popped off `M2H_bloodmeal`, and added to `E2I_human` at times $b_i + \tau_{LEP}$. Simulation then proceeds as before.